



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	1	

<b>COBGDB - The GnuCOBOL TUI DEBUGGER / ANIMATOR</b>
<i>HOW TO USE COBGDB</i>



## Table of Contents

1.Introduction.....	2
1.1.Installing the debugger on Windows.....	3
1.2.Compile and Debug GnuCOBOL programs.....	4
1.3.Main Commands .....	5
2.Tutorial - Sample Debugging Session .....	6
2.1.Help Command.....	8
2.2.Run Command.....	10
2.3.Step Command .....	12
2.4.Go Command .....	14
2.5.Show Command .....	15
2.6.Variable Command.....	17
2.6.1.Enter subCommand.....	17
2.6.2.Edit subCommand.....	19
2.6.3.Return subCommand.....	20
2.7.Step Command .....	21
2.8.Pop-up Variable windows .....	24
2.9.File Command .....	25
2.10.Run Command.....	28
2.11.Quit Command .....	29
3.Document Change Log.....	30

DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	2	

## 1. Introduction

COBGDB is a TUI (Text User Interface) application, programmed in C, designed to assist in animate and debugging GnuCOBOL TUI code **using GDB**.

The project is hosted at <https://github.com/marcsosduma/cobgdb>

***Very important: you don't need to know GDB and all its many commands  
(<https://www.sourceware.org/gdb/>).***


***COBGDB has its own interface that is very simple to use  
and is responsible for interfacing the underlying GDB which is the real debug and animate engine  
but operates practically in a transparent and invisible way for the user.***

The COBGDB application is based on the extension for Visual Studio Code (VSCode) created by Oleg Kunitsyn, which can be found on GitHub: <https://github.com/OlegKunitsyn/gnucobol-debug>.  
warning: COBGDB is still currently under development.

***At <https://github.com/marcsosduma/cobgdb> in the Windows subdirectory,  
the executable program cobgdb.exe for this operating system is available and ready to use.***

To compile COBGDB from C source code on Windows, you can use MinGW.

The Makefile is configured to generate the program **cobgdb . exe** for both Windows and Linux.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	3	

## 1.1. Installing the debugger on Windows

On Windows, just download **cobgdb.exe** from following folder:  
<https://github.com/marcsosduma/cobgdb/tree/main/windows>.

As an example you can put **cobgdb.exe** into the "bin" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler **cobc.exe** is located)

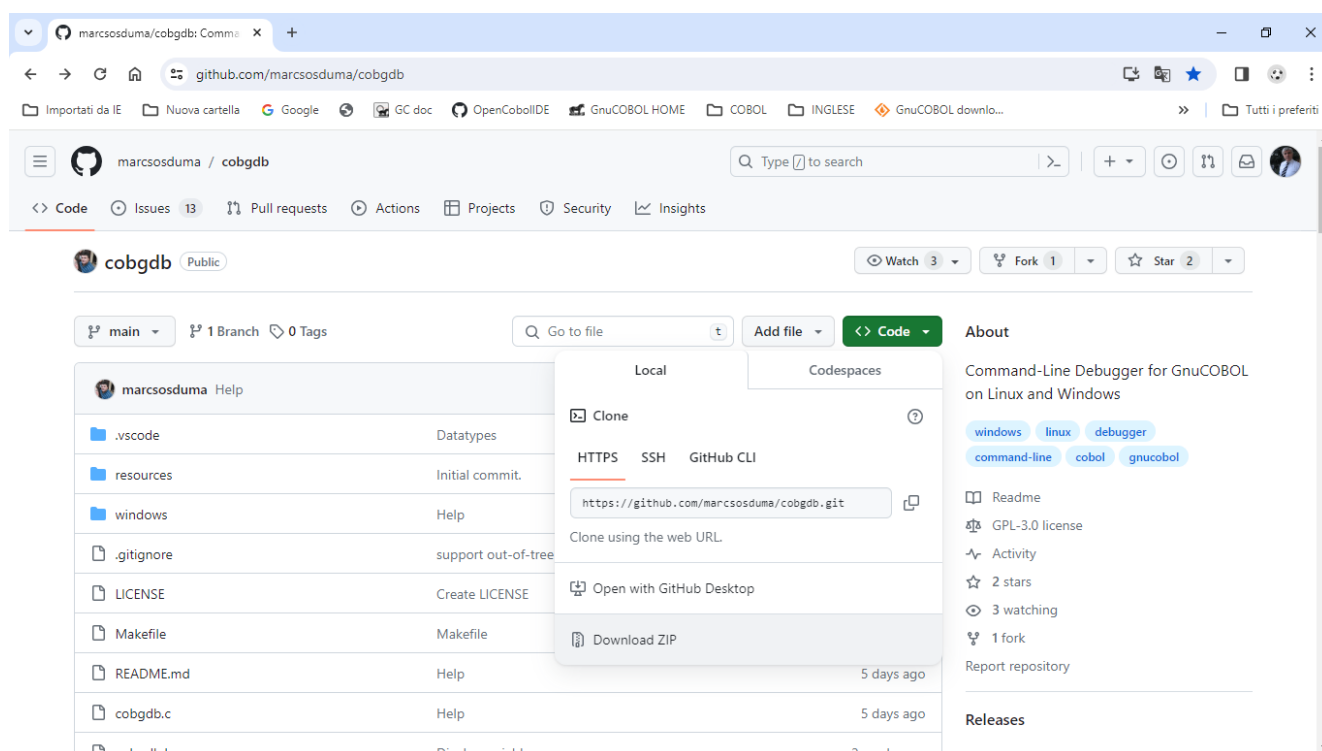
or


first install MinGW (Minimalist GNU for Windows).

Then execute the make ('mingw32-make' for Windows) command to compile the code from C source.

Note: if you have security problems downloading an .exe file then you can try download the entire repository with the following github button **<> Code v --> Download ZIP**

Then unzip the file and copy cobgdb.exe to the "bin" folder of your GnuCOBOL installation (the same folder where the GnuCOBOL compiler cobc.exe is located)



DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	4	

## 1.2. Compile and Debug GnuCOBOL programs

Compile and run a debugging session of the sample program using the following command:

**cobgdb customer.cob -x -lpdcurses**

Source code of customer.cob used also for following tutorial is at:

<https://github.com/marcsosduma/cobgdb/tree/main/windows>

Note: '-lpdcurses' is an instance of an argument that can be indirectly passed to 'cobc' by 'cobgdb,' even if it is not used by 'cobgdb' itself.

or, other example for cobc parameters , use : **cobgdb customer.cob -x -Tcustomer.txt** .(-T creates a compilation list output into **customer.txt** file )

COBGDB takes one or more programs with COB or CBL extension as parameters and runs the GnuCOBOL compiler with the following format:

**cobc -g -fsource-location -ftraceall -v -free -O0 -x prog.cob prog2.cob ...**

To debug multiple programs, use COBGDB with the following syntax:

**cobgdb prog.cob subprog1.cob subprog2.cob ...**


This will create a single prog.exe executable.

You can run GDB/GDBSERVER remotely using the "A" (Attach) key.

COBGDB will prompt you to provide the server and port in the format server: port or the PID of the application.


**Example:**

- localhost: 5555
- 9112

DOCUMENT CODE	MODULE: xxxxxxxxxx	<b>USING COBGDB FOR GnuCOBOL</b>	PAGE	<b>GnuCOBOL</b>
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	5	

### 1.3. Main Commands

Cmd		Description	
<b>?</b>	Help	Show the HELP window and text	
<b>B</b>	Breakpoint	Toggle (Set or Unset) a breakpoint at a specific line of the Procedure Division code.	
<b>R</b>	Run	Runs the program <u>from the first Cobol statement</u> until a breakpoint is encountered. Always use this command to start a debugging session.	
<b>C</b>	Cursor	Runs the program until it reaches the selected line.	
<b>N</b>	Next	Runs the program until the next line but does not enter a subroutine executed by CALL or PERFORM.	
<b>S</b>	Step	Runs the program until the next line. If needed it enter a subroutine executed by CALL or PERFORM.	
<b>G</b>	Go	Continues the program execution until it encounters a stopping point: a breakpoint, end of the program, or the return from a subroutine (PERFORM / CALL).	
<b>J</b>	Jump	Ask for a line number and Runs the program until it reaches that line.	
<b>V</b>	Variables	Displays a new window with the set of variables for the running program. From this window you can also change the value of variables	
<b>H</b>	sHow	Shows a window with the values of variables on the selected line.	
<b>F</b>	File	When cobgdb is executed with more than one program, allows selecting one of those source file for debugging commands.	
<b>A</b>	Attach	Attach to GDBSERVER or Application PID.	
<b>Q</b>	Quit	Quits (Ends) the debugging session and the program.	

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	6	

## 2. Tutorial - Sample Debugging Session

Following tutorial is on a Windows 10 platform using following version of GnuCOBOL:

```
cobc (GnuCOBOL) 3.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Keisuke Nishida, Roger While, Ron Norman, Simon Sobisch, Edward Hart
Built Jul 28 2023 16:07:38
Packaged Jul 28 2023 16:58:47 UTC
C version (MinGW) "13.1.0"
```

Downloaded from <https://www.arnoldtrembley.com/GnuCOBOL.htm>


===== Version 3.2 =====

**GnuCOBOL 3.2 (28Jul2023) MSYS2 64-bit [GC32M-BDB-x64.7z](#)** -- MSYS2 64-bit GnuCOBOL 3.2

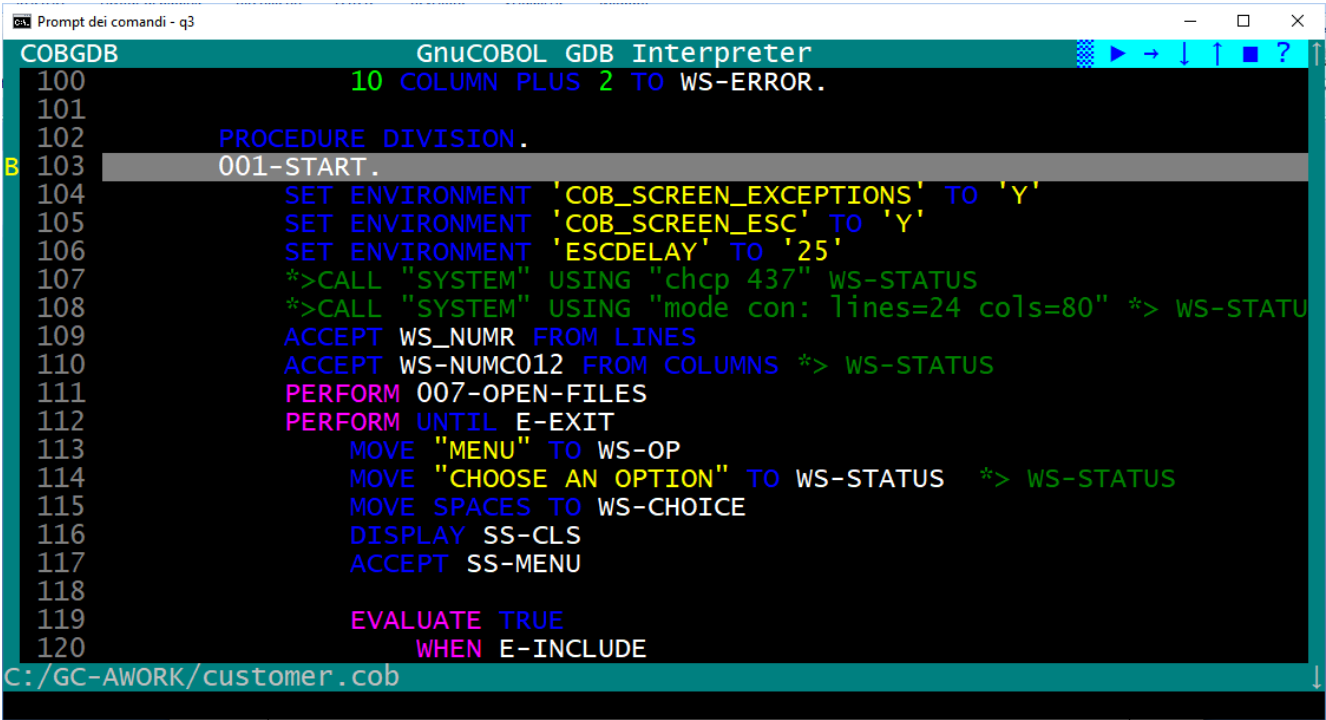
Final release **with full debugging support**. (95.4 Megabytes).

**GnuCOBOL 3.2 (28Jul2023) MSYS2 32-bit [GC32M-BDB-x32.7z](#)** -- MSYS2 32-bit GnuCOBOL 3.2

Final release **with full debugging support**. (96.1 Megabytes).

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	7	

After executing **cobgdb customer.cob -x -lpcurses** the application automatically insert a Breakpoint at first executable program statement of PROCEDURE DIVISION (see the **B** symbol at left of line 103 in this sample) and displays following screen:




You can scroll the source code window with cursor keys UP and DOWN, PG UP and PG DOWN or with mouse wheel or with mouse left click on the right scroll bar. Use cursor RIGHT and cursor LEFT to scroll horizontally,

In the upper right window corner there is a "button bar" where you can find some buttons (symbols):


> → ↓ ↑ ■ ?

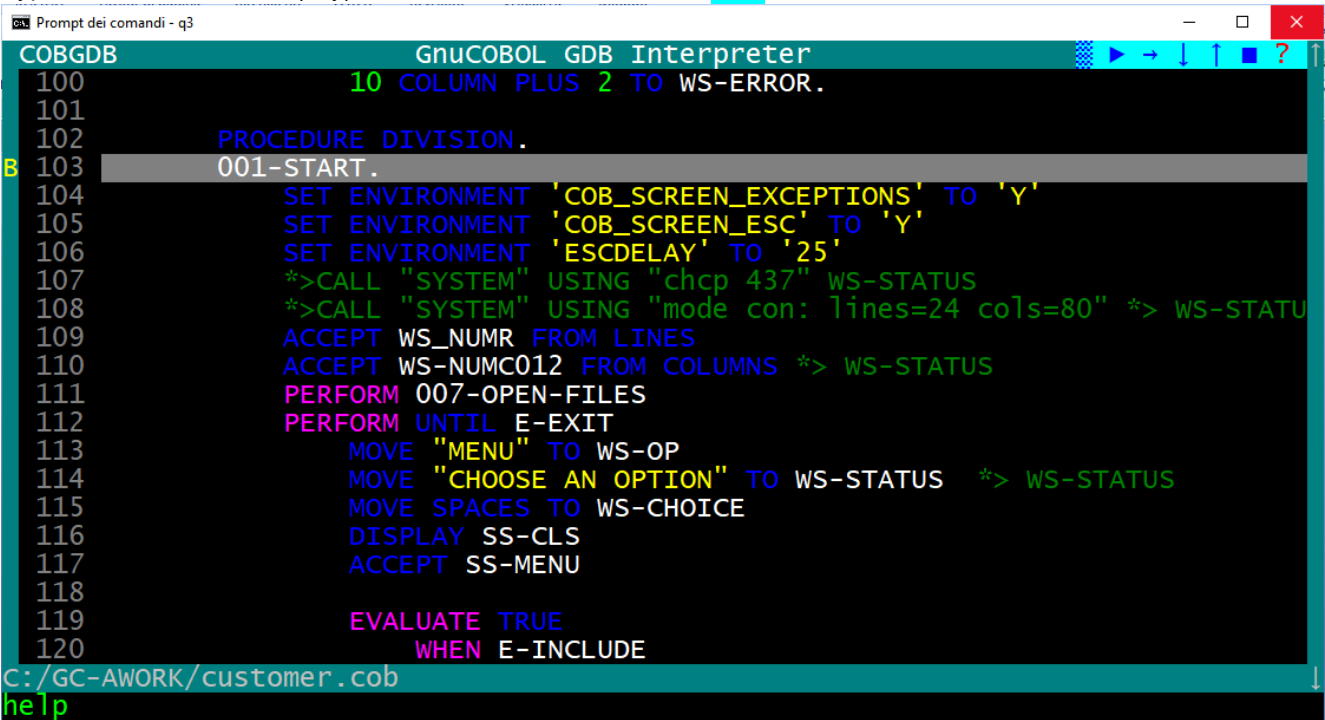
>	=	<b>Run</b>	command
→	=	<b>Next</b>	command
↓	=	<b>Step</b>	command
↑	=	<b>Go</b>	command
■	=	<b>Quit</b>	command
?	=	<b>Help</b>	command

when you hover over one of these commands, you get the corresponding command description (like a tooltip) displayed at the bottom left of the screen.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	8	

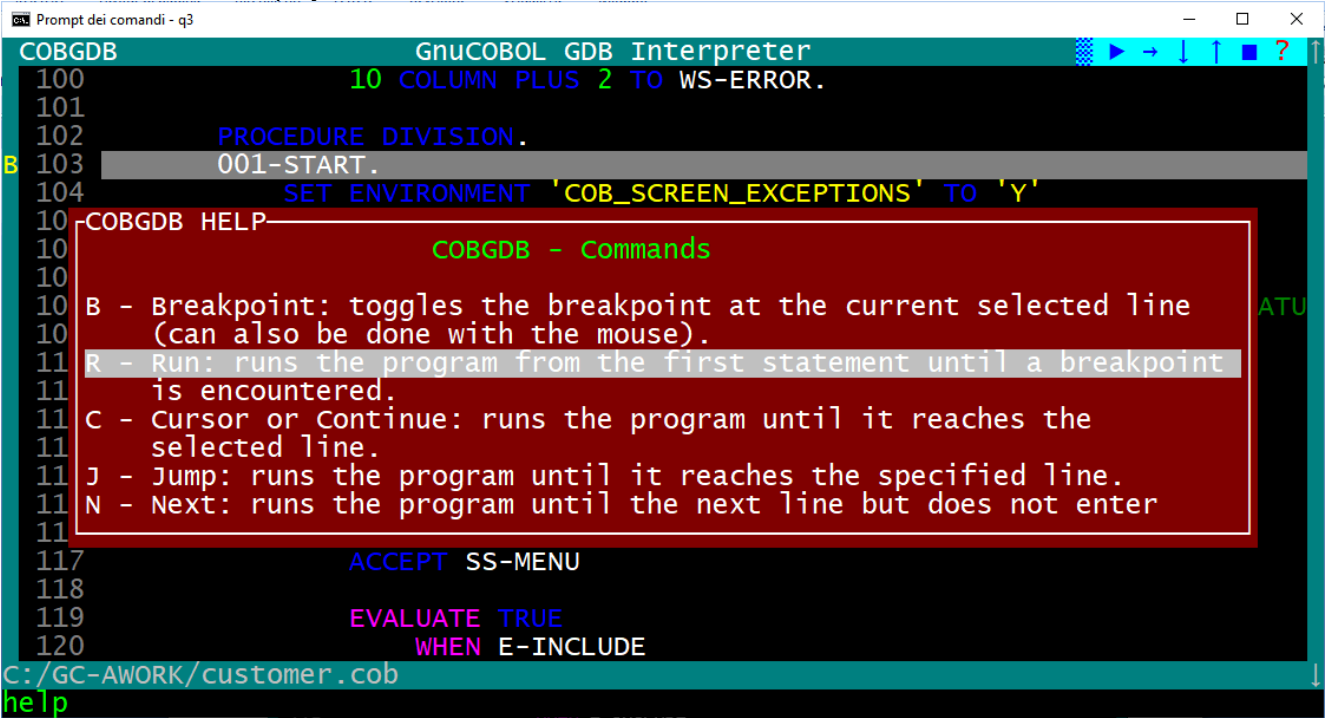
### 2.1. Help Command

Type ? HELP command (key) or left click with mouse on the  button:



The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window. The title bar is "Prompt dei comandi - q3". The window has a menu bar with "COBGDB" and "GnuCOBOL GDB Interpreter". The main area displays COBOL code with line numbers 100 to 120. Line 100 is "10 COLUMN PLUS 2 TO WS-ERROR.". Line 101 is blank. Line 102 is "PROCEDURE DIVISION.". Line 103 is "001-START.". Line 104 is "SET ENVIRONMENT 'COB\_SCREEN\_EXCEPTIONS' TO 'Y'". Line 105 is "SET ENVIRONMENT 'COB\_SCREEN\_ESC' TO 'Y'". Line 106 is "SET ENVIRONMENT 'ESCDELAY' TO '25'". Line 107 is ">CALL 'SYSTEM' USING 'chcp 437' WS-STATUS". Line 108 is ">CALL 'SYSTEM' USING 'mode con: lines=24 cols=80' > WS-STATUS". Line 109 is "ACCEPT WS\_NUMR FROM LINES". Line 110 is "ACCEPT WS\_NUMC012 FROM COLUMNS > WS-STATUS". Line 111 is "PERFORM 007-OPEN-FILES". Line 112 is "PERFORM UNTIL E-EXIT". Line 113 is "MOVE 'MENU' TO WS-OP". Line 114 is "MOVE 'CHOOSE AN OPTION' TO WS-STATUS > WS-STATUS". Line 115 is "MOVE SPACES TO WS-CHOICE". Line 116 is "DISPLAY SS-CLS". Line 117 is "ACCEPT SS-MENU". Line 118 is blank. Line 119 is "EVALUATE TRUE". Line 120 is "WHEN E-INCLUDE". The status bar at the bottom shows "C:/GC-AWORK/customer.cob" and "help".

the HELP window is displayed




The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window with the HELP window displayed. The title bar is "Prompt dei comandi - q3". The window has a menu bar with "COBGDB" and "GnuCOBOL GDB Interpreter". The main area displays COBOL code with line numbers 100 to 120. Line 100 is "10 COLUMN PLUS 2 TO WS-ERROR.". Line 101 is blank. Line 102 is "PROCEDURE DIVISION.". Line 103 is "001-START.". Line 104 is "SET ENVIRONMENT 'COB\_SCREEN\_EXCEPTIONS' TO 'Y'". Line 105 is "SET ENVIRONMENT 'COB\_SCREEN\_ESC' TO 'Y'". Line 106 is "SET ENVIRONMENT 'ESCDELAY' TO '25'". Line 107 is ">CALL 'SYSTEM' USING 'chcp 437' WS-STATUS". Line 108 is ">CALL 'SYSTEM' USING 'mode con: lines=24 cols=80' > WS-STATUS". Line 109 is "ACCEPT WS\_NUMR FROM LINES". Line 110 is "ACCEPT WS\_NUMC012 FROM COLUMNS > WS-STATUS". Line 111 is "PERFORM 007-OPEN-FILES". Line 112 is "PERFORM UNTIL E-EXIT". Line 113 is "MOVE 'MENU' TO WS-OP". Line 114 is "MOVE 'CHOOSE AN OPTION' TO WS-STATUS > WS-STATUS". Line 115 is "MOVE SPACES TO WS-CHOICE". Line 116 is "DISPLAY SS-CLS". Line 117 is "ACCEPT SS-MENU". Line 118 is blank. Line 119 is "EVALUATE TRUE". Line 120 is "WHEN E-INCLUDE". The status bar at the bottom shows "C:/GC-AWORK/customer.cob" and "help". A red box highlights the HELP window content:


```
COBGDB HELP
COBGDB - commands
B - Breakpoint: toggles the breakpoint at the current selected line
  (can also be done with the mouse).
R - Run: runs the program from the first statement until a breakpoint
  is encountered.
C - Cursor or Continue: runs the program until it reaches the
  selected line.
J - Jump: runs the program until it reaches the specified line.
N - Next: runs the program until the next line but does not enter
```

scroll the Help window with cursor keys UP and DOWN or mouse wheel.



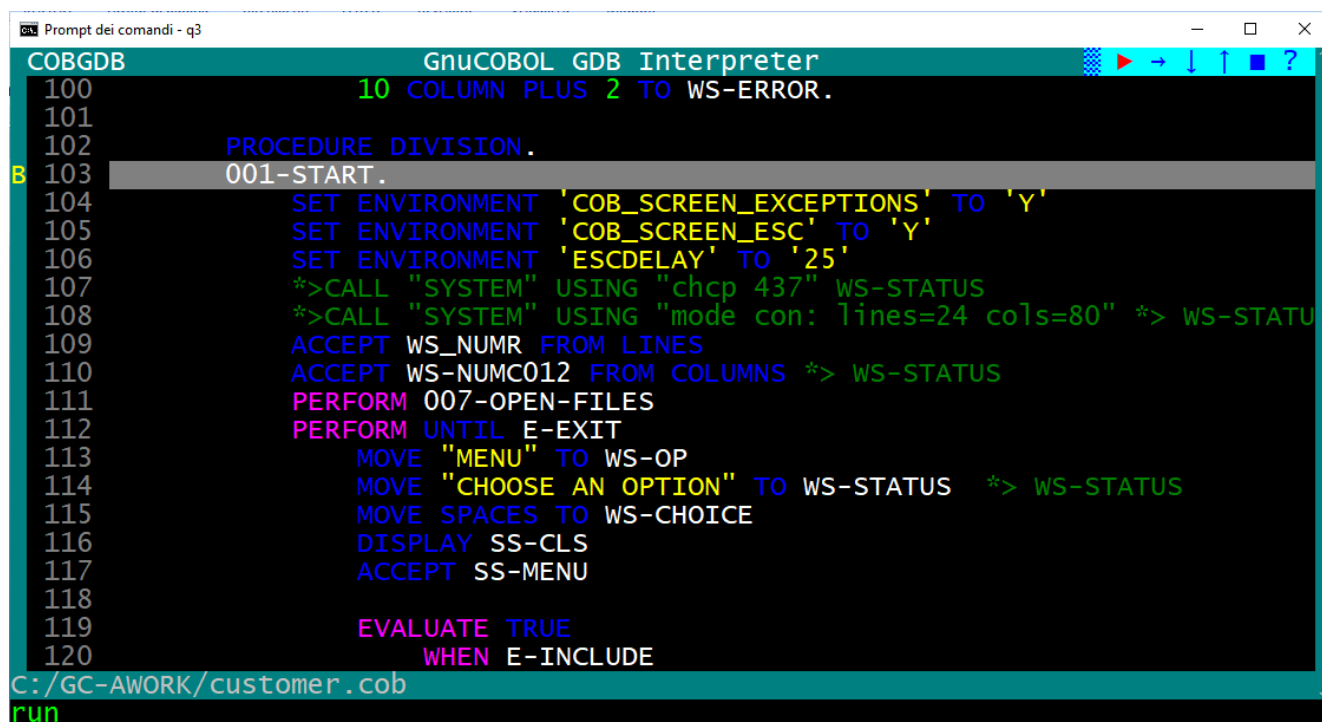
DOCUMENT CODE	MODULE: xxxxxxxxxx	<b>USING COBGDB FOR GnuCOBOL</b>	PAGE	<b>GnuCOBOL</b>
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	9	

Use ESC or Enter or left click to exit from this HELP window and return to debugging session.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	10	

## 2.2. Run Command

To start executing the program and the debugging session from first program statement you always must use the "R" command (key) or left click with mouse on the  **Run** button




```

COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
run

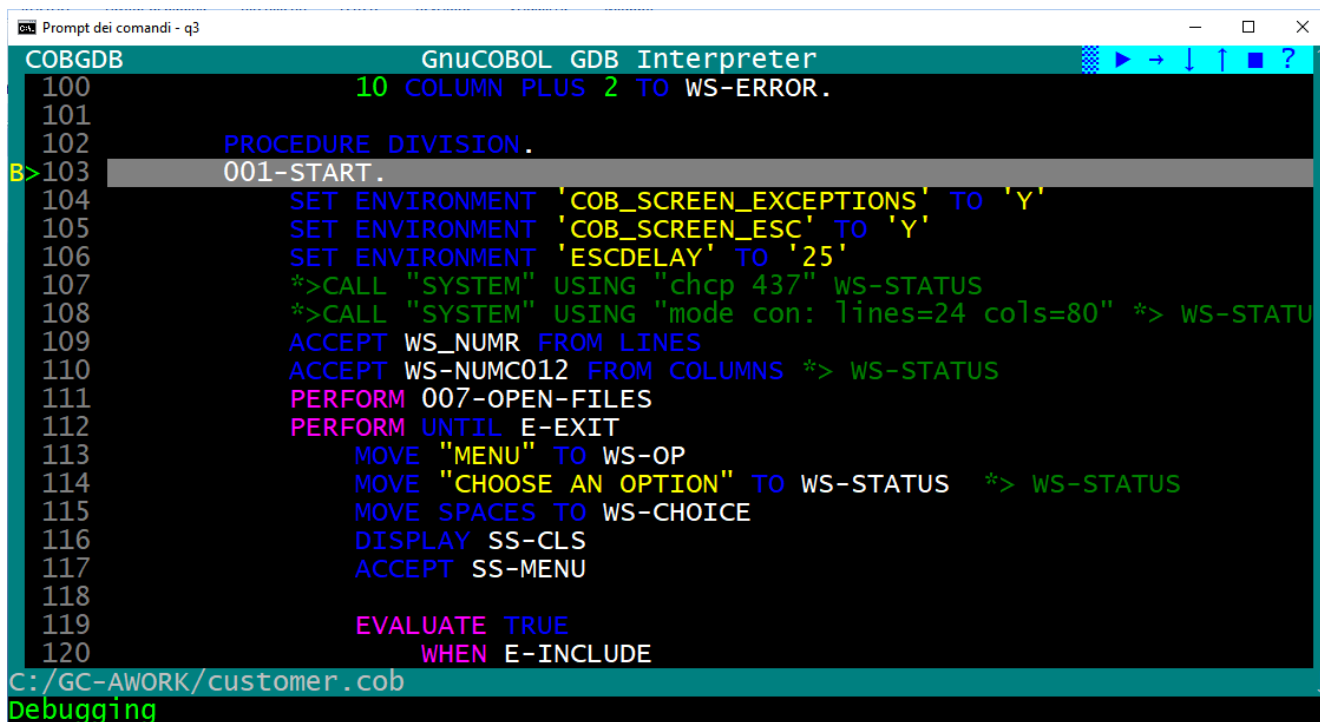
```

cobgdb opens the program terminal window (the application will run in this separate window. ).



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	11	

go back to the COBGDB screen, and you see a  green symbol on the left of statement where initial Breakpoint is present (in our example is at line 103 **B> 103**):




```

COBGDB                                GnuCOBOL GDB Interpreter
100                                10 COLUMN PLUS 2 TO WS-ERROR.
101
102                                PROCEDURE DIVISION.
B>103                               001-START.
104                                SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105                                SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106                                SET ENVIRONMENT 'ESCDELAY' TO '25'
107                                *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108                                *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109                                ACCEPT WS_NUMR FROM LINES
110                                ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
111                                PERFORM 007-OPEN-FILES
112                                PERFORM UNTIL E-EXIT
113                                    MOVE "MENU" TO WS-OP
114                                    MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115                                    MOVE SPACES TO WS-CHOICE
116                                    DISPLAY SS-CLS
117                                    ACCEPT SS-MENU
118
119                                    EVALUATE TRUE
120                                    WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging


```

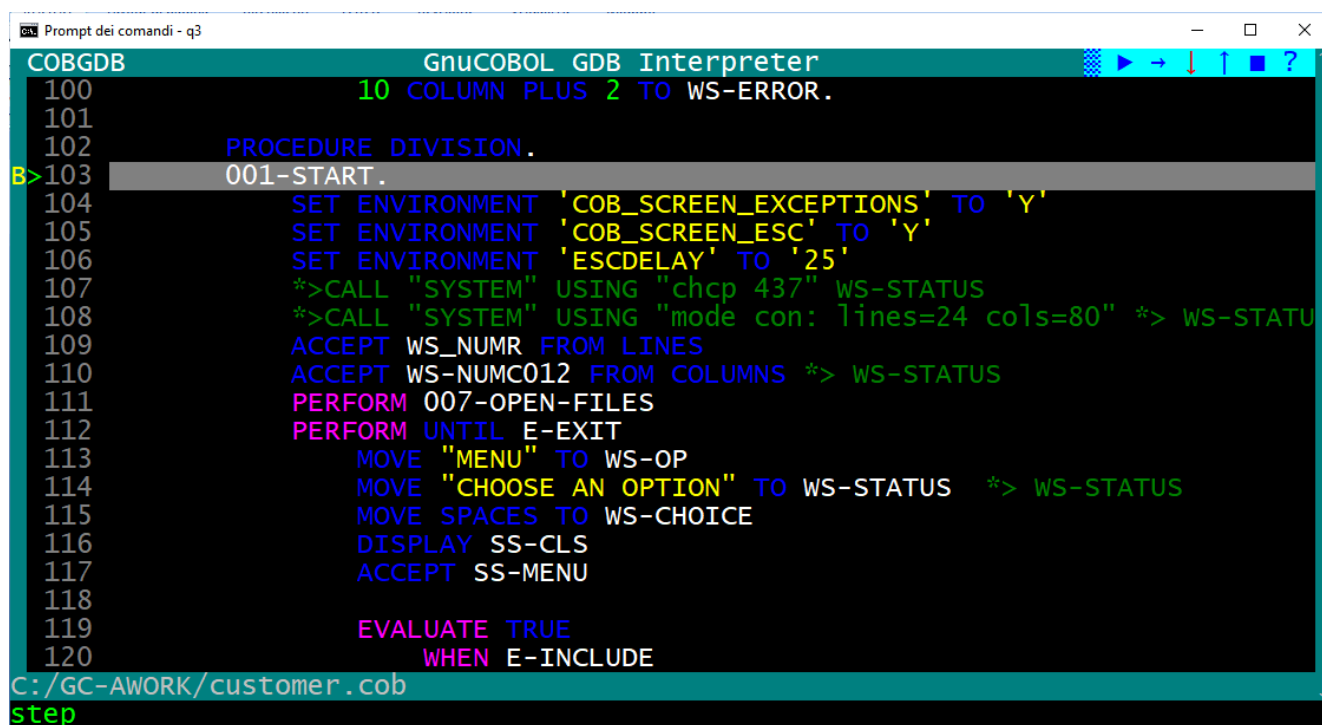
From that moment on, you can use all the commands (keys) or corresponding buttons to "animate" and debug the application, example: **"S"** (Step), **"N"** (Next) and **"G"** (Go) and so on.

During the source code animation, the debugger automatically shows some pop-up windows with variables content from the line in execution.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	12	

### 2.3. Step Command


Proceed with **S** (Step) command or left click with mouse the  button:.

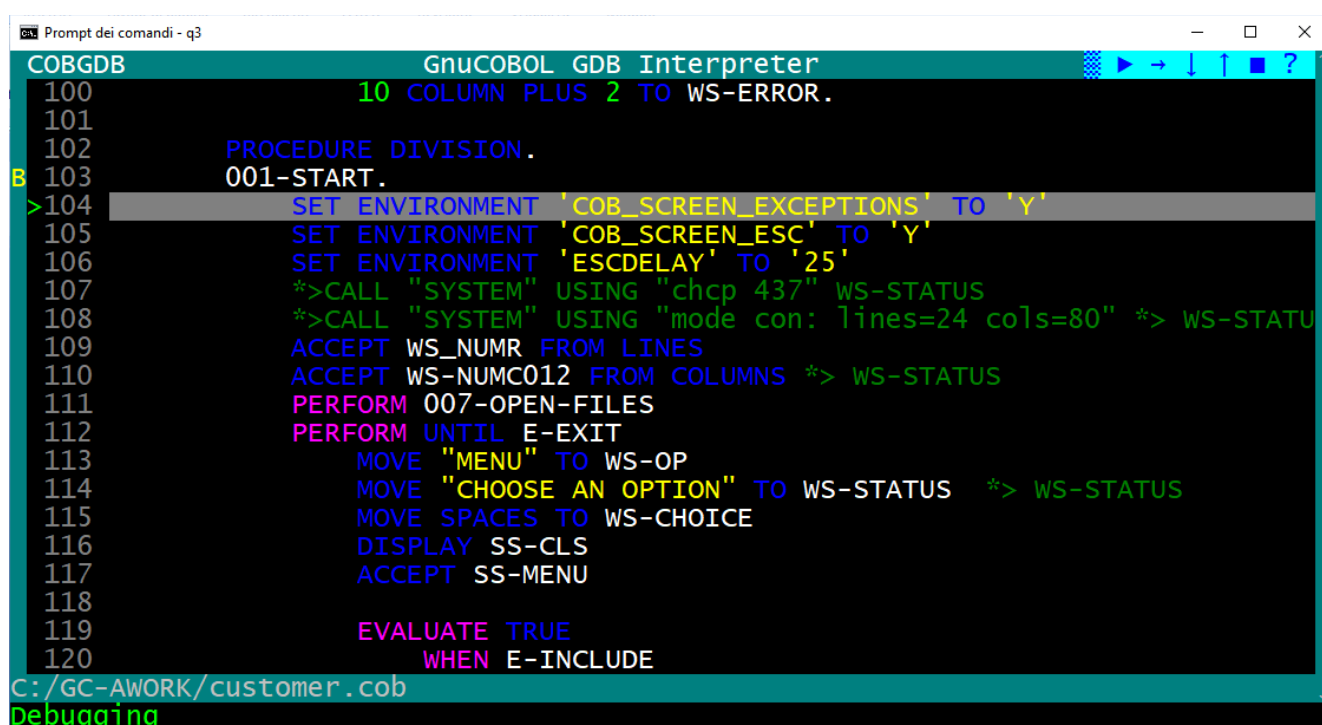


```

COBGDB          GnuCOBOL GDB Interpreter
100              10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
B>103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step

```


the  green symbol now is on the following line 104:



```

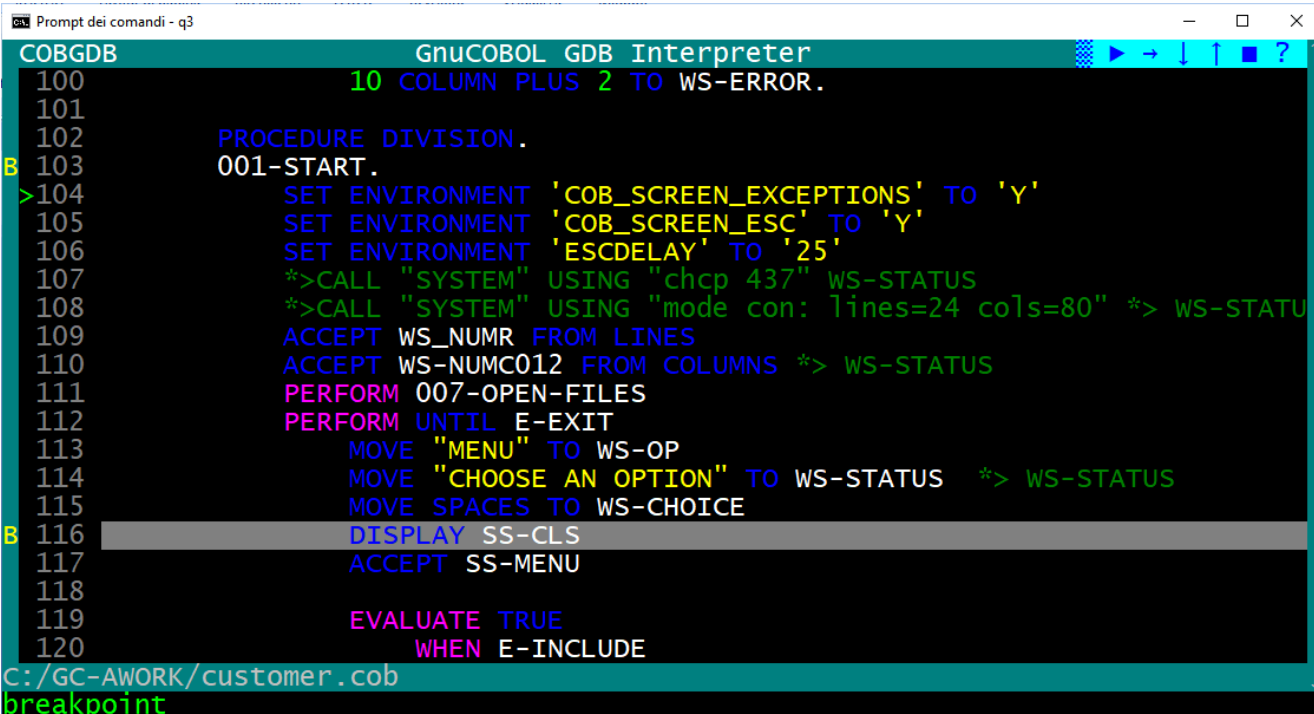
COBGDB          GnuCOBOL GDB Interpreter
100              10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
B>103      001-START.
>104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	13	

Now you can proceed with S command or N command or as an example:


- Scroll with cursor down to select line 116 of Procedure Division and type "B" (to set a Breakpoint), (you also can simply click with mouse left button on the 116 row number)
- The application displays a "B" on the left of the line (type B again - or re-click - when you want to delete the Breakpoint, not do that at this moment)





```

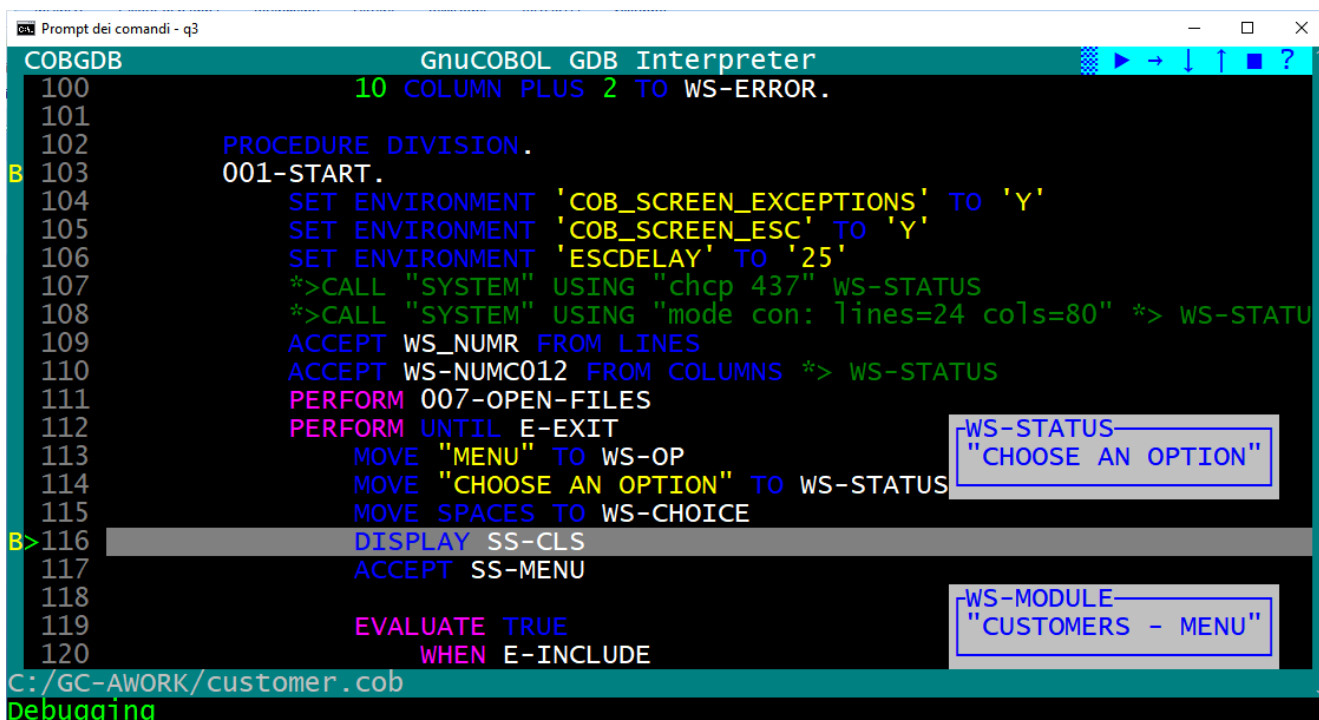
COBGDB GnuCOBOL GDB Interpreter
100 10 COLUMN PLUS 2 TO WS-ERROR.
101
102 PROCEDURE DIVISION.
103 001-START.
104 SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105 SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106 SET ENVIRONMENT 'ESCDELAY' TO '25'
107 *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108 *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109 ACCEPT WS_NUMR FROM LINES
110 ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111 PERFORM 007-OPEN-FILES
112 PERFORM UNTIL E-EXIT
113     MOVE "MENU" TO WS-OP
114     MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115     MOVE SPACES TO WS-CHOICE
116 B  DISPLAY SS-CLS
117     ACCEPT SS-MENU
118
119     EVALUATE TRUE
120     WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
breakpoint

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	14	

## 2.4. Go Command


Type **G** (Go) or left click with mouse the  button to execute the program until a B Breakpoint is detected: the system reach the second breakpoint at line 116 and displays a green  symbol to the left of the line to be executed, see following screen:



```

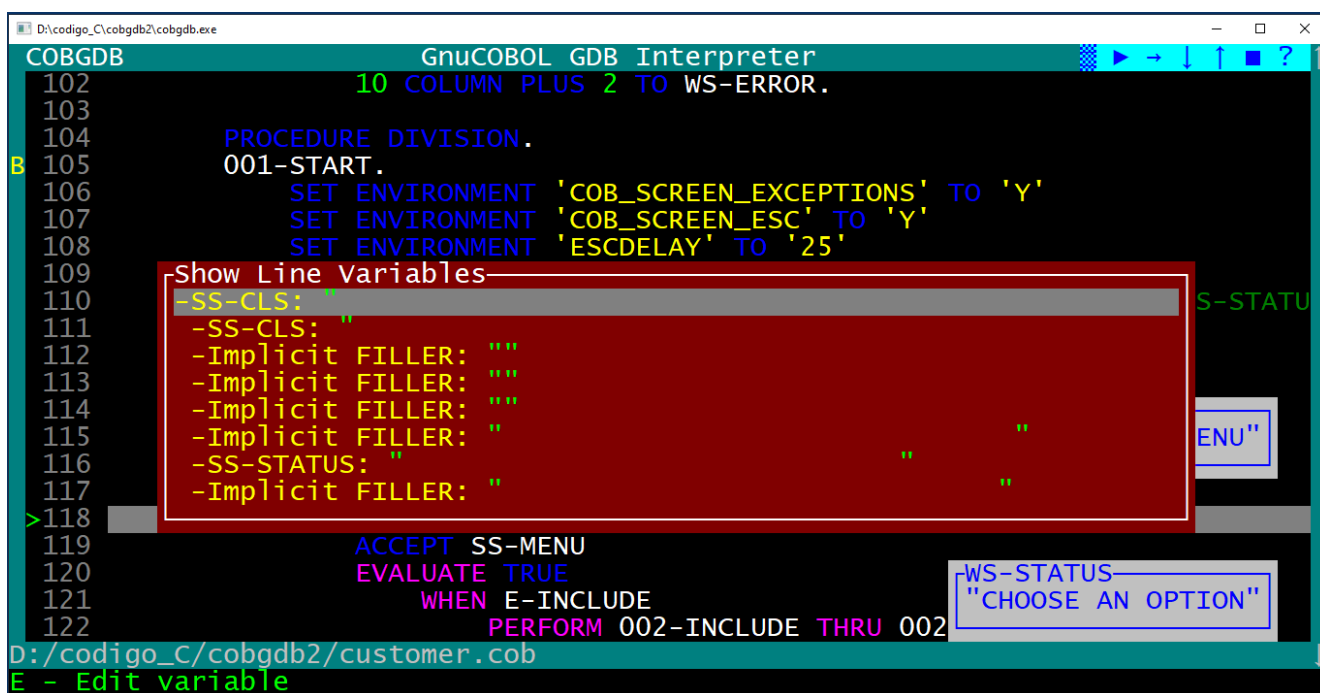
COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116      B> 116      DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE
WS-STATUS
"CHOOSE AN OPTION"
WS-MODULE
"CUSTOMERS - MENU"
C:/GC-AWORK/customer.cob
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	15	

## 2.5. Show Command

typing the 'H' command (key) allows you to view the variables on the highlighted line.



```

COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      -SS-CLS: "
111      -SS-CLS: "
112      -Implicit FILLER: ""
113      -Implicit FILLER: ""
114      -Implicit FILLER: ""
115      -Implicit FILLER: ""
116      -SS-STATUS: ""
117      -Implicit FILLER: ""
118
119      ACCEPT SS-MENU
120      EVALUATE TRUE
121          WHEN E-INCLUDE
122              PERFORM 002-INCLUDE THRU 002
  
```

Show Line Variables

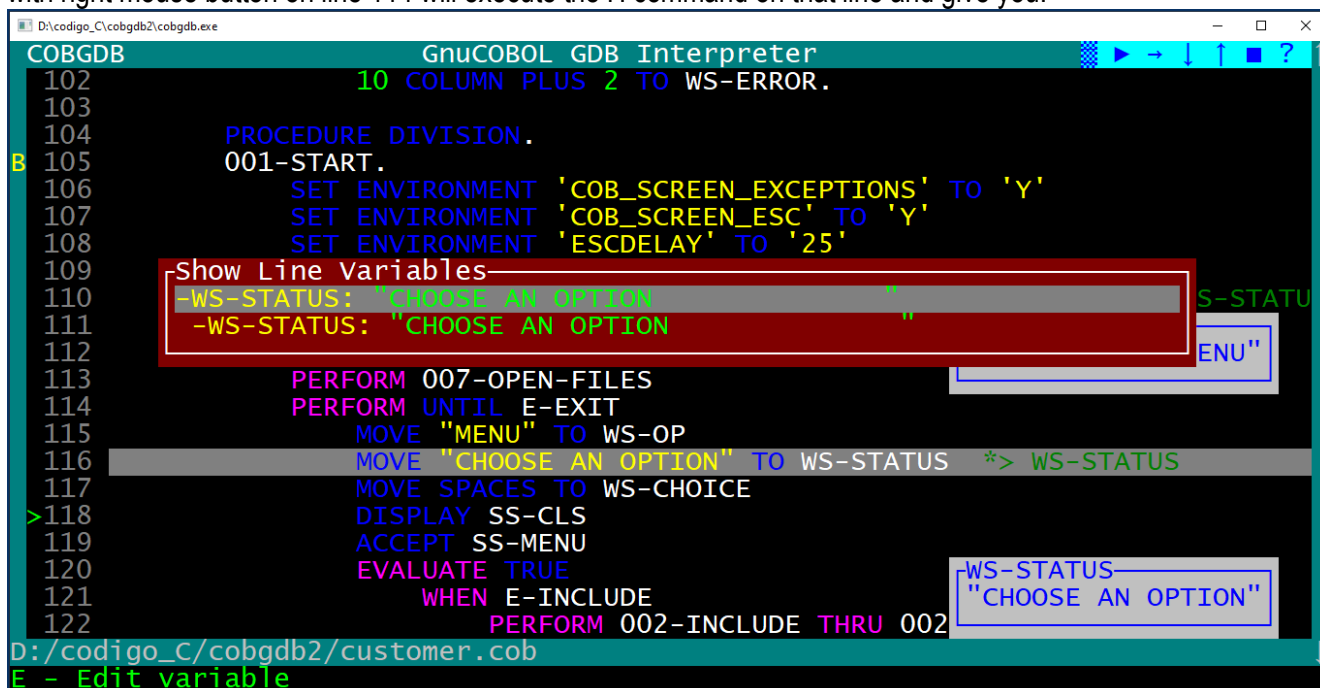
- SS-CLS: "
- SS-CLS: "
- Implicit FILLER: ""
- Implicit FILLER: ""
- Implicit FILLER: ""
- Implicit FILLER: ""
- SS-STATUS: ""
- Implicit FILLER: ""

WS-STATUS: "CHOOSE AN OPTION"

D:/codigo\_C/cobgdb2/customer.cob

E - Edit variable

Note: to display the content of variables you can also click with right mouse button on a source line, example click with right mouse button on line 114 will execute the H command on that line and give you:



```

COBGDB GnuCOBOL GDB Interpreter
102      10 COLUMN PLUS 2 TO WS-ERROR.
103
104      PROCEDURE DIVISION.
105      001-START.
106          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108          SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110      -WS-STATUS: "CHOOSE AN OPTION"
111      -WS-STATUS: "CHOOSE AN OPTION"
112
113      PERFORM 007-OPEN-FILES
114      PERFORM UNTIL E-EXIT
115          MOVE "MENU" TO WS-OP
116          MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
117          MOVE SPACES TO WS-CHOICE
118      DISPLAY SS-CLS
119      ACCEPT SS-MENU
120      EVALUATE TRUE
121          WHEN E-INCLUDE
122              PERFORM 002-INCLUDE THRU 002
  
```


Show Line Variables

- WS-STATUS: "CHOOSE AN OPTION"
- WS-STATUS: "CHOOSE AN OPTION"

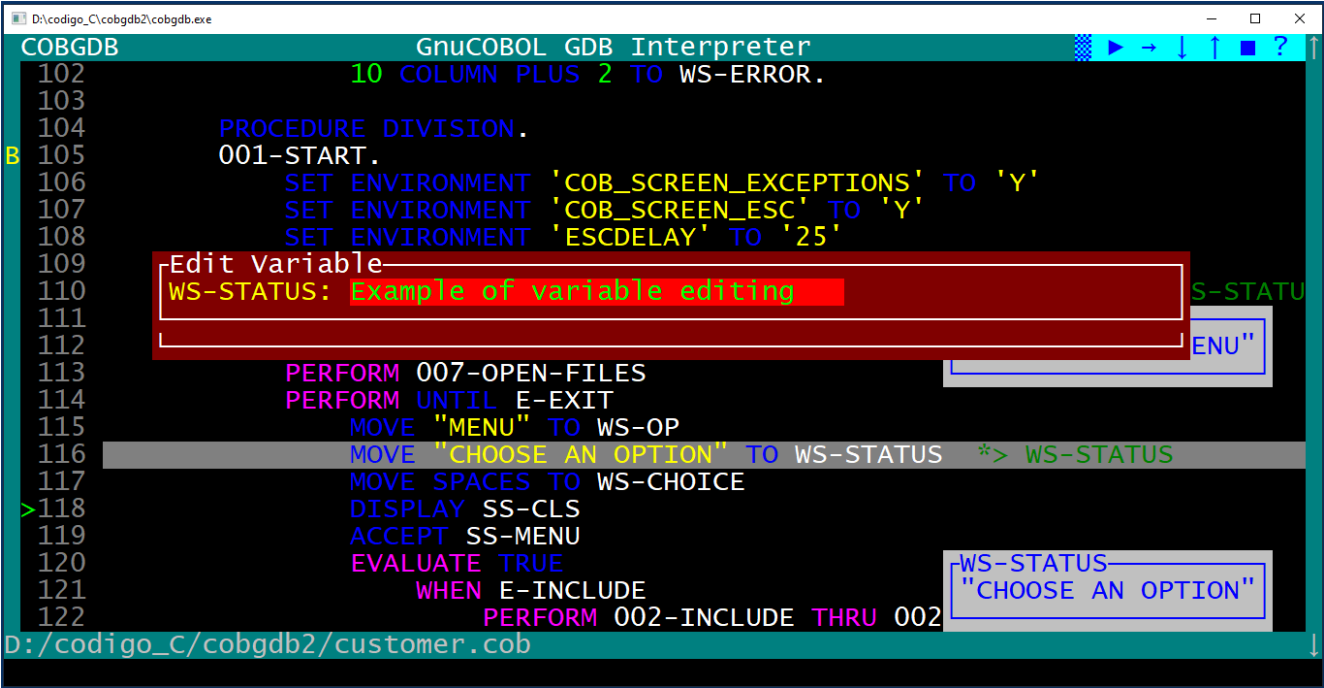
WS-STATUS: "CHOOSE AN OPTION"

D:/codigo\_C/cobgdb2/customer.cob

E - Edit variable

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	16	

Scroll with cursor key UP and DOWN (or with mouse wheel). The key **E** can be used to edit the content of the highlighted variable.

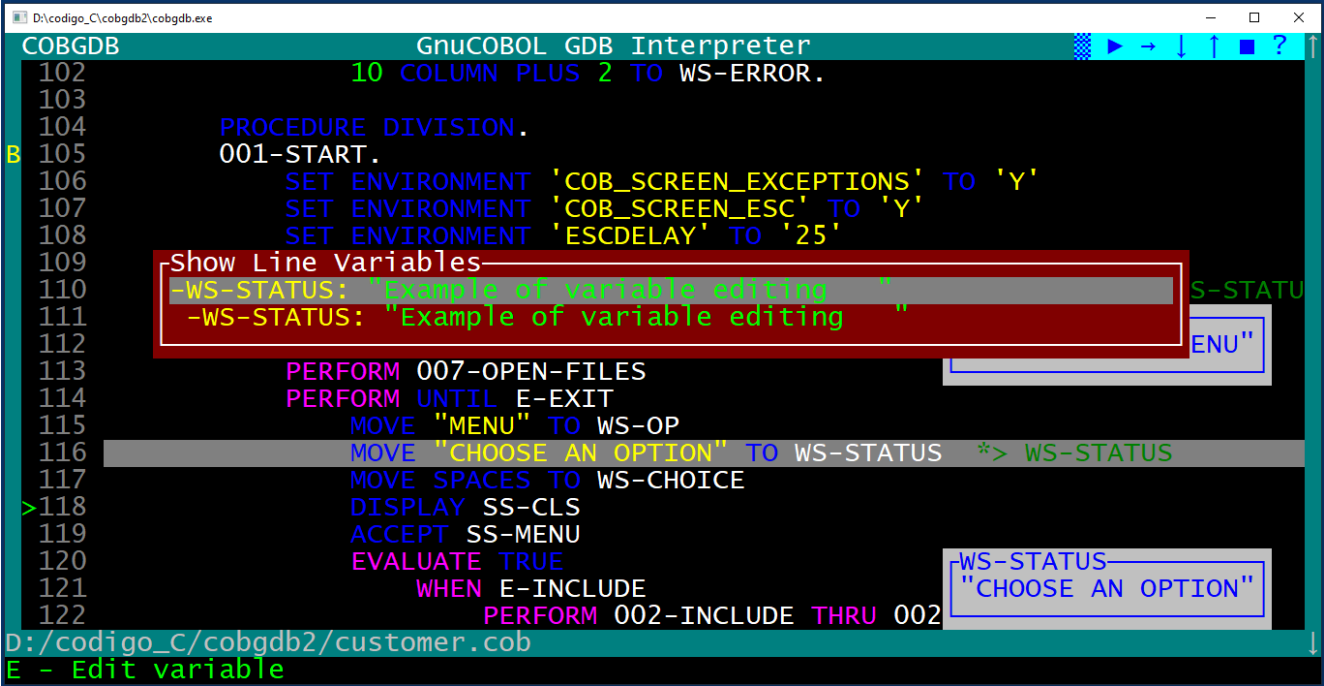


```

COBGDB                               GnuCOBOL GDB Interpreter
102                                10 COLUMN PLUS 2 TO WS-ERROR.
103
104        PROCEDURE DIVISION.
105    001-START.
106        SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107        SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108        SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110    Edit Variable
111    WS-STATUS: Example of variable editing
112
113        PERFORM 007-OPEN-FILES
114        PERFORM UNTIL E-EXIT
115            MOVE "MENU" TO WS-OP
116            MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
117            MOVE SPACES TO WS-CHOICE
118        DISPLAY SS-CLS
119        ACCEPT SS-MENU
120        EVALUATE TRUE
121            WHEN E-INCLUDE
122                PERFORM 002-INCLUDE THRU 002
WS-STATUS: "CHOOSE AN OPTION"
D:/codigo_c/cobgdb2/customer.cob

```

Resulting:




```

COBGDB                               GnuCOBOL GDB Interpreter
102                                10 COLUMN PLUS 2 TO WS-ERROR.
103
104        PROCEDURE DIVISION.
105    001-START.
106        SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
107        SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
108        SET ENVIRONMENT 'ESCDELAY' TO '25'
109
110    Show Line Variables
111    -WS-STATUS: "Example of variable editing"
112    -WS-STATUS: "Example of variable editing"
113
114        PERFORM 007-OPEN-FILES
115        PERFORM UNTIL E-EXIT
116            MOVE "MENU" TO WS-OP
117            MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
118            MOVE SPACES TO WS-CHOICE
119        DISPLAY SS-CLS
120        ACCEPT SS-MENU
121        EVALUATE TRUE
122            WHEN E-INCLUDE
123                PERFORM 002-INCLUDE THRU 002
WS-STATUS: "CHOOSE AN OPTION"
D:/codigo_c/cobgdb2/customer.cob
E - Edit variable

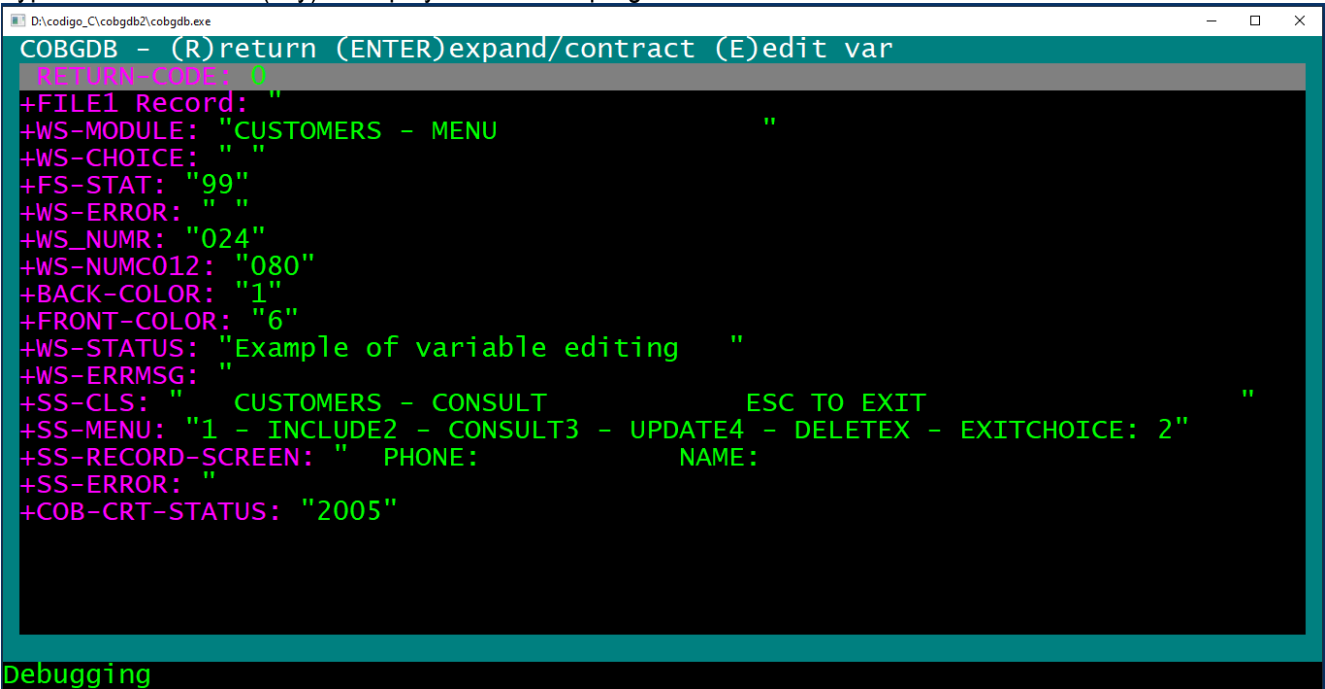
```



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	17	

### 2.6. Variable Command

Type the **V** command (key) to display the list of all program variables:



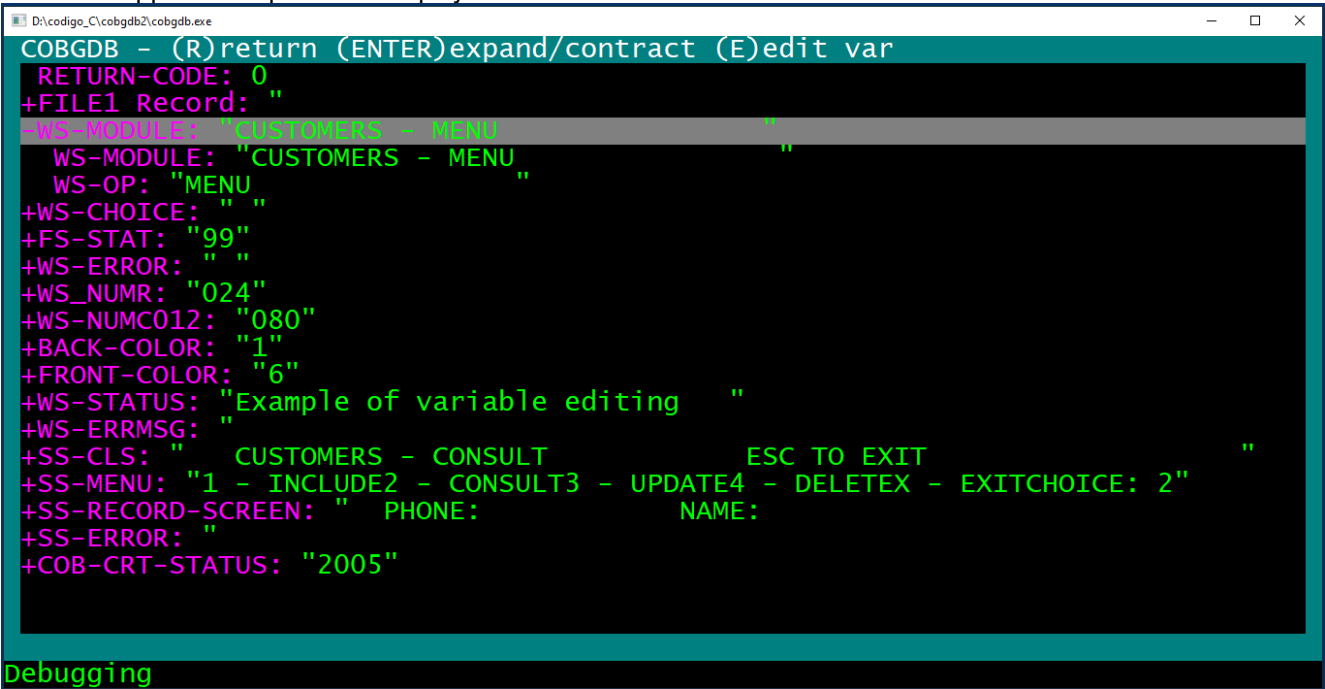
```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
+WS-MODULE: "CUSTOMERS - MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: "
+SS-CLS: "CUSTOMERS - CONSULT" ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: "
+COB-CRT-STATUS: "2005"
Debugging

```

#### 2.6.1.Enter subCommand


Scroll with cursor key UP and DOWN (or with mouse wheel) in this list to the variable WS-MODULE and type Enter: the application opens and displays its subfields WS-MODULE and WS-OP



```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
+WS-MODULE: "CUSTOMERS - MENU"
  WS-MODULE: "CUSTOMERS - MENU"
  WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: "
+SS-CLS: "CUSTOMERS - CONSULT" ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "PHONE: NAME:"
+SS-ERROR: "
+COB-CRT-STATUS: "2005"
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	18	


select WS-MODULE subfield

```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "CUSTOMERS - MENU"
WS-MODULE: "CUSTOMERS - MENU"
WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing"
+WS-ERRMSG: "
+SS-CLS: "      CUSTOMERS - CONSULT          ESC TO EXIT"
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETEx - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "  PHONE:          NAME:"
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

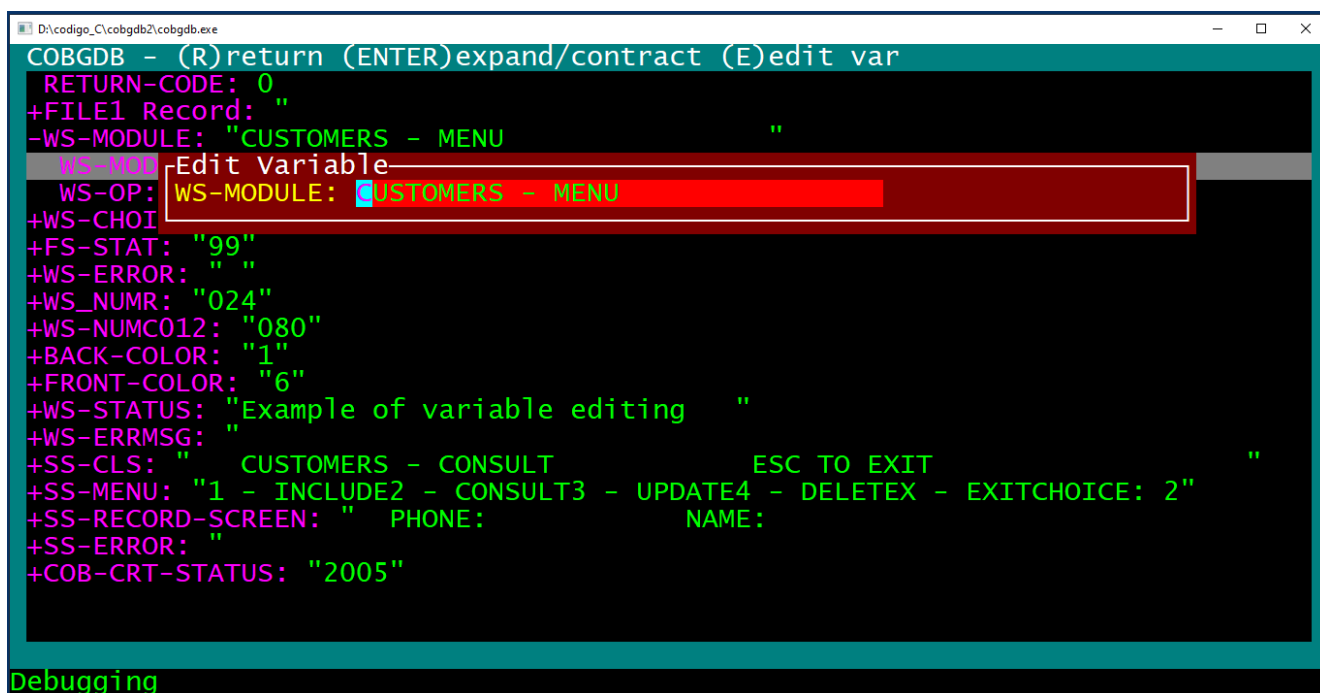
Debugging

```

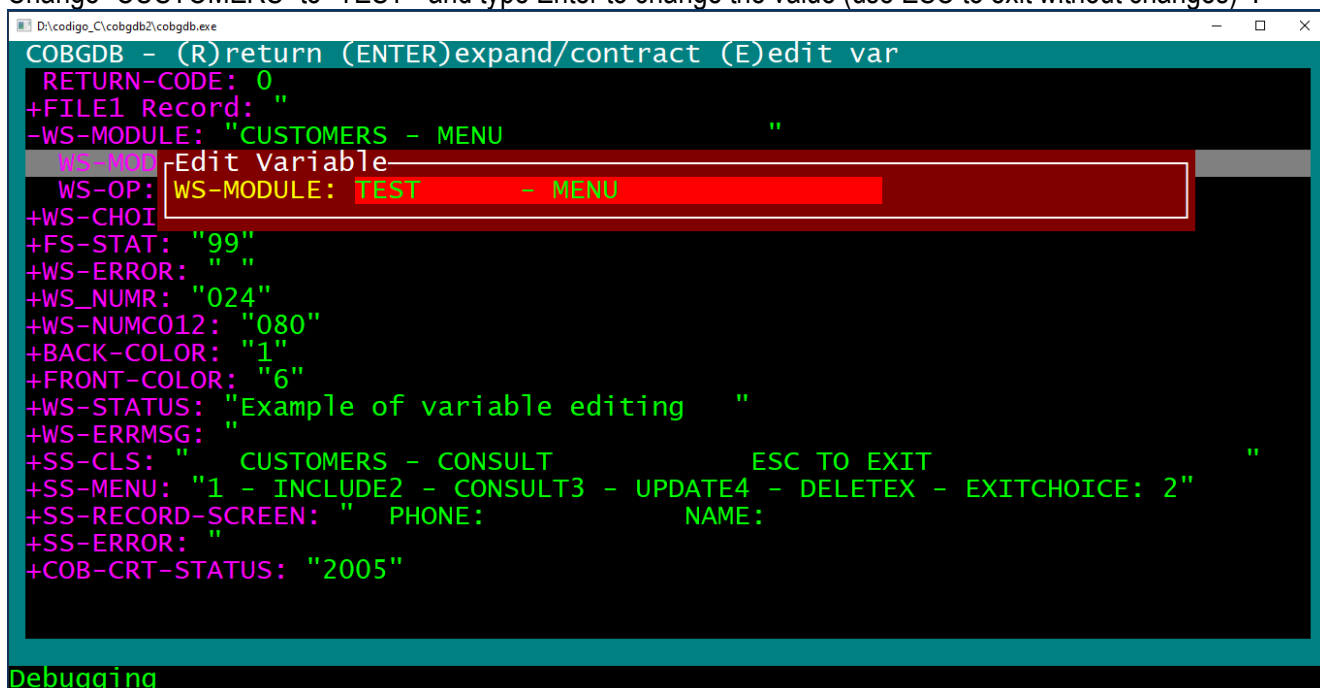
DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	19	


### 2.6.2.Edit subCommand

Now you can select WS-MODULE subfield whit cursor DOWN or mouse wheel and type "E" (Edit)  
COBGDB shows a Edit Variable window:



Change "CUSTOMERS" to "TEST" and type Enter to change the value (use ESC to exit without changes) :

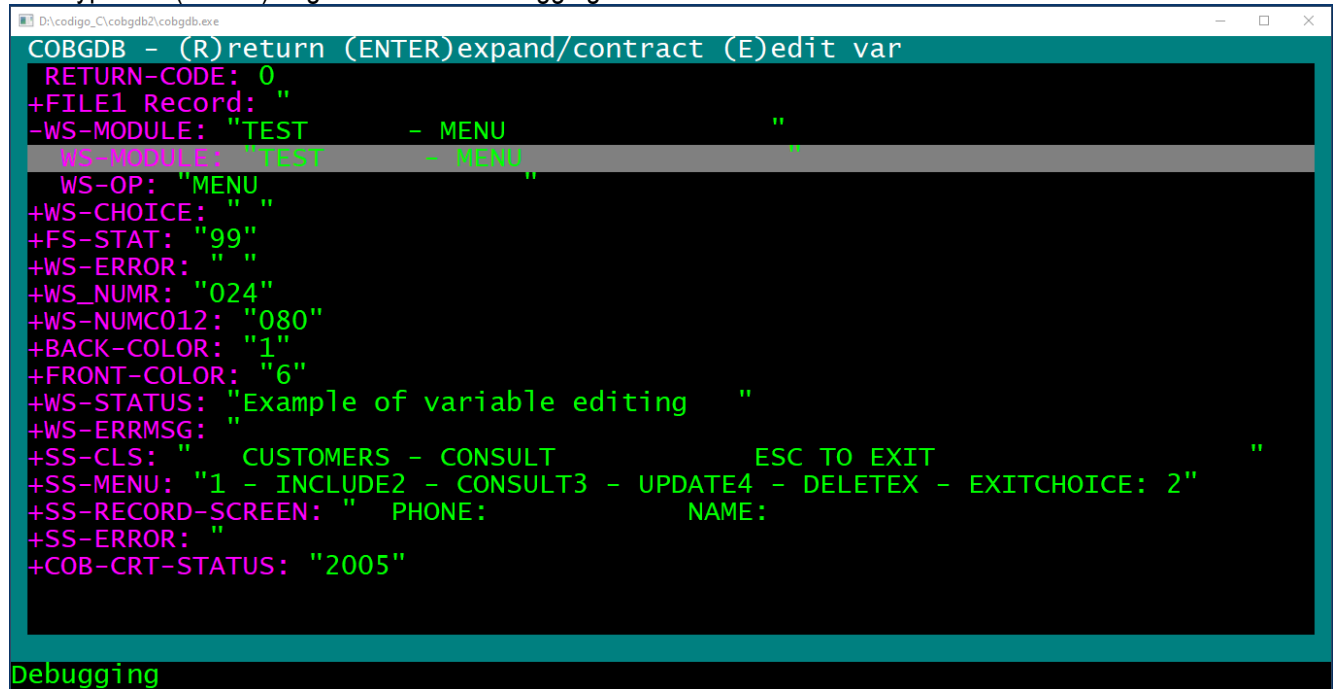


DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	20	

### 2.6.3.Return subCommand

WS-Module has new value.

Now type "R" (Return) to go back to the debugging session:




```

D:\codigo_C\cobgdb2\cobgdb.exe
COBGDB - (R)return (ENTER)expand/contract (E)edit var
RETURN-CODE: 0
+FILE1 Record: "
-WS-MODULE: "TEST      - MENU      "
WS-MODULE: "TEST      - MENU      "
WS-OP: "MENU"
+WS-CHOICE: " "
+FS-STAT: "99"
+WS-ERROR: " "
+WS_NUMR: "024"
+WS-NUMC012: "080"
+BACK-COLOR: "1"
+FRONT-COLOR: "6"
+WS-STATUS: "Example of variable editing  "
+WS-ERRMSG: "
+SS-CLS: "      CUSTOMERS - CONSULT      ESC TO EXIT      "
+SS-MENU: "1 - INCLUDE2 - CONSULT3 - UPDATE4 - DELETX - EXITCHOICE: 2"
+SS-RECORD-SCREEN: "  PHONE:      NAME:
+SS-ERROR: "
+COB-CRT-STATUS: "2005"

Debugging

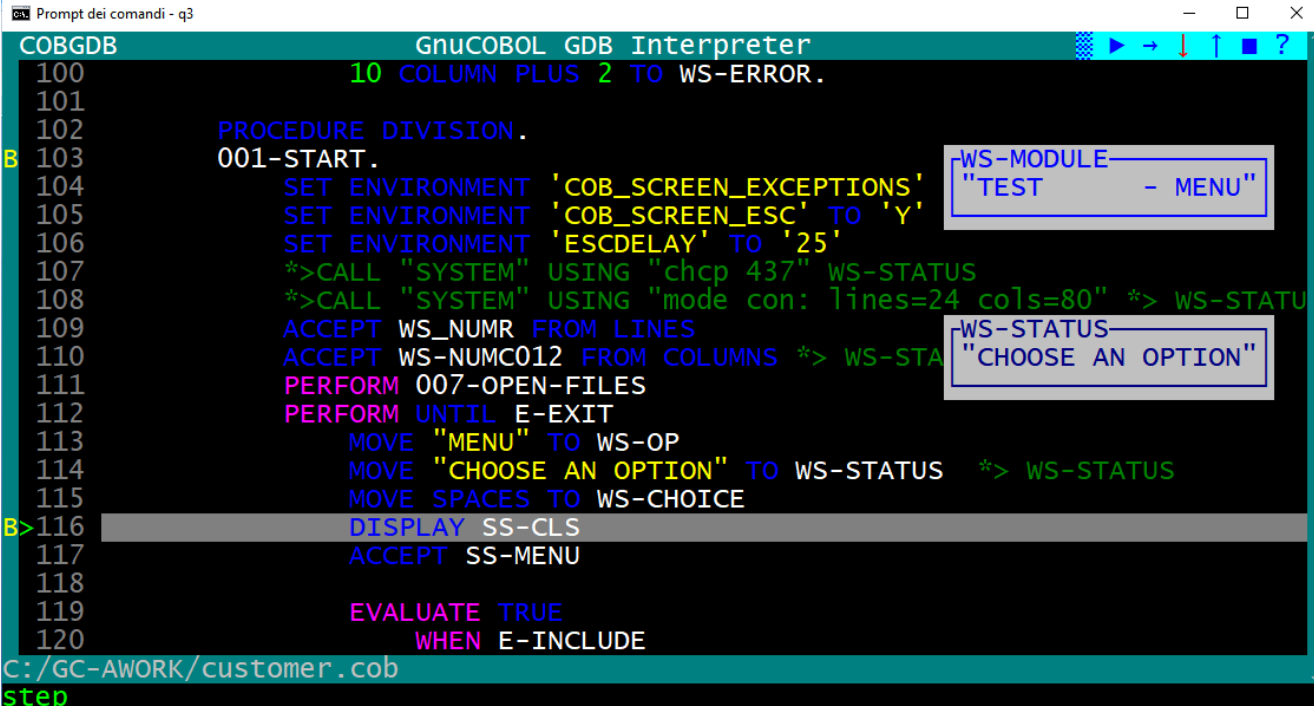
```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	21	

## 2.7. Step Command

now you are back in the main debugging window:

Type **S** (Step) command or left click with mouse the  button to execute the DISPLAY statement at line 116



```


COBGDB GnuCOBOL GDB Interpreter
100      10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116      B> 116 DISPLAY SS-CLS
117          ACCEPT SS-MENU
118
119          EVALUATE TRUE
120              WHEN E-INCLUDE
  
```


C:/GC-AWORK/customer.cob  
step

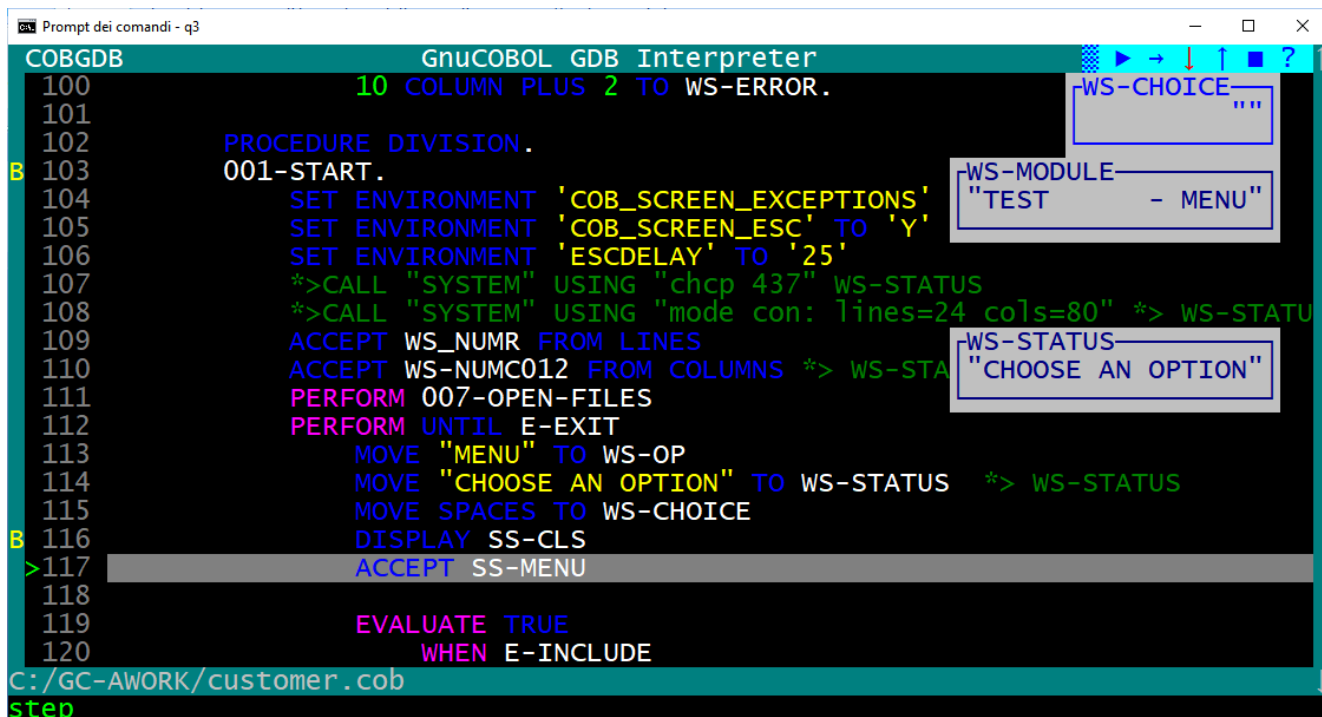
in the other application window you can see the result of DISPLAY statement



go back to debugging window and now the ACCEPT statement will be executed with S command

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	22	

Type **S** (Step) command or left click with mouse the  button again to execute the ACCEPT statement at line 117:



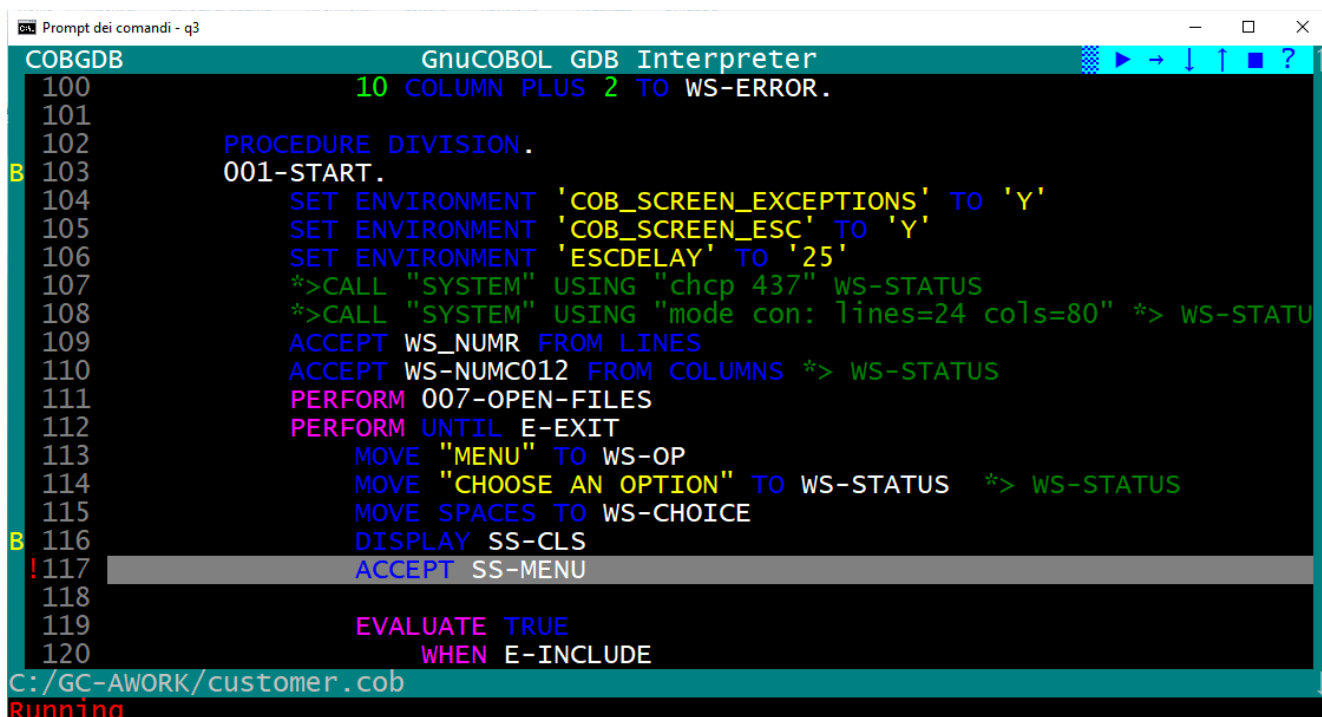
```

COBGDB          GnuCOBOL GDB Interpreter
100              10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103  B 001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116  B  DISPLAY SS-CLS
117  >  ACCEPT SS-MENU
118
119          EVALUATE TRUE
120              WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
step

```

A red **!** quotation mark appears on the line **! 117**.


This means that application is running and a user action is required at application window.



```

COBGDB          GnuCOBOL GDB Interpreter
100              10 COLUMN PLUS 2 TO WS-ERROR.
101
102      PROCEDURE DIVISION.
103  B 001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110          ACCEPT WS_NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116  B  DISPLAY SS-CLS
117  ! ACCEPT SS-MENU
118
119          EVALUATE TRUE
120              WHEN E-INCLUDE
C:/GC-AWORK/customer.cob
Running

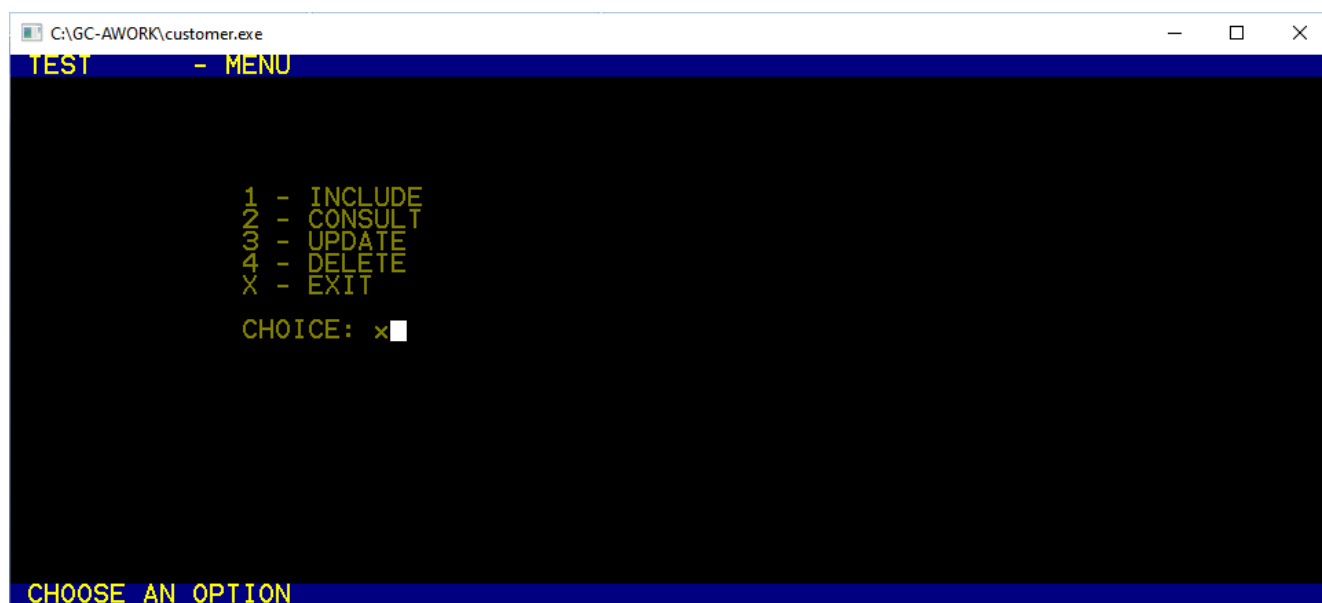
```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	23	

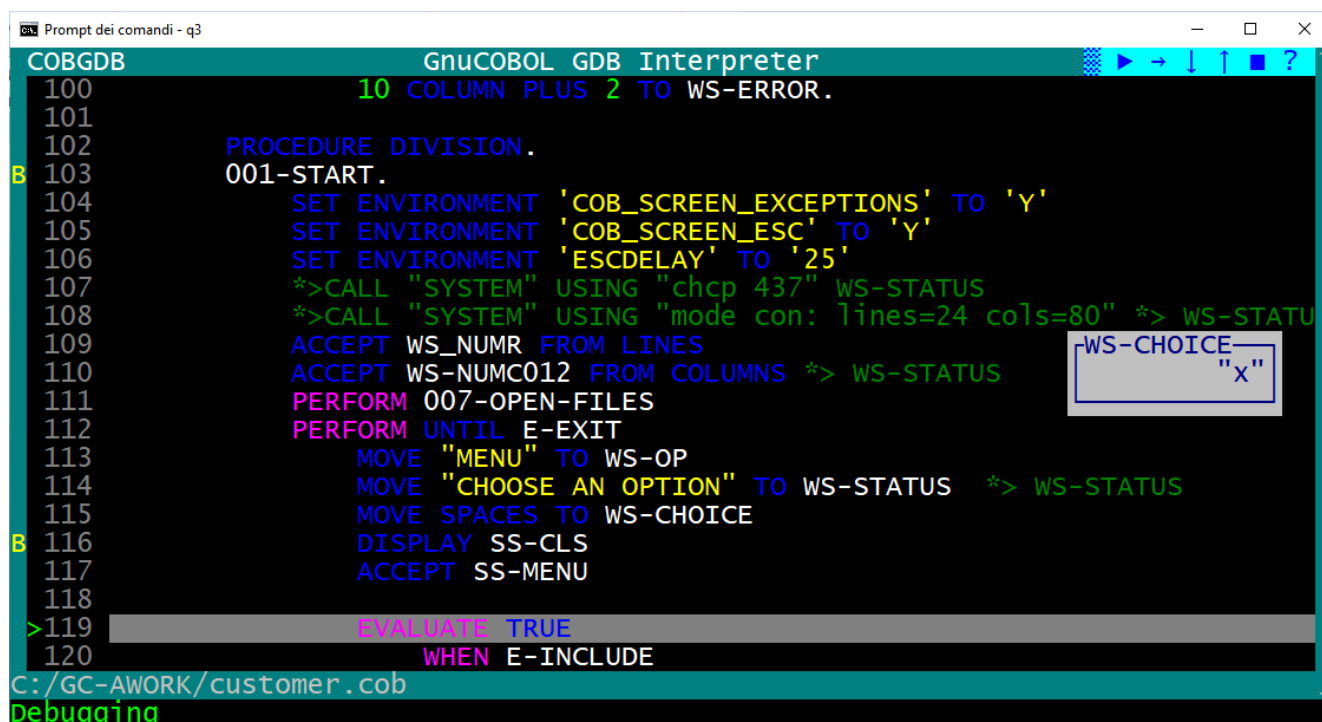
The application is running in a separate window.


The 'Accept' command switches the focus to application windows.

After a user action on application screen (ex type the "X" choice and Enter) it is necessary to click again on the 'debugger' window to continue debugging.



go back to debugger window: the ACCEPT statement has been executed:



DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	24	

## 2.8. Pop-up Variable windows

During a debugging session COBGDB shows variable content.

Blue frame and values : variables of executing cobol statement

Black frame and values : variables of last executed cobol statement.

Sample:




```

COBGDB GnuCOBOL GDB Interpreter
262 continue after 0.5 seconds
263
264 continue.
265 DisplayCoverEx. exit.
266
267 TetraminoDisplay.
268 perform varying wRow from 1 by 1 until wRow > 4
269 perform varying wCol from 1 by 1 until wCol > 4
270 compute wLinD = wBaseLin + wRow
271 compute wColD = wBaseCol + wCol
272 if wBlockEle (wRow wCol) = 'X'
B> 273 display wChar at line wLinD column wColD :BC0: blue :FC0: wCol
274 end-if
275 end-perform
276 end-perform
277 continue.
278 TetraminoDisplayEx. exit.
279
280 TetraminoSet
281 move spa
282 evaluate
C:/GC-AWORK/GC99TETRIS.COB
go

```



DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	25	

## 2.9. File Command

To show this command we use following sample:

**cobgdb sample.cbl subsample.cbl subsubsample.cbl -x -lpcurses**

where sample.cbl is the main program; it calls

--> subsample.cbl; it calls

--> subsubsample.cbl

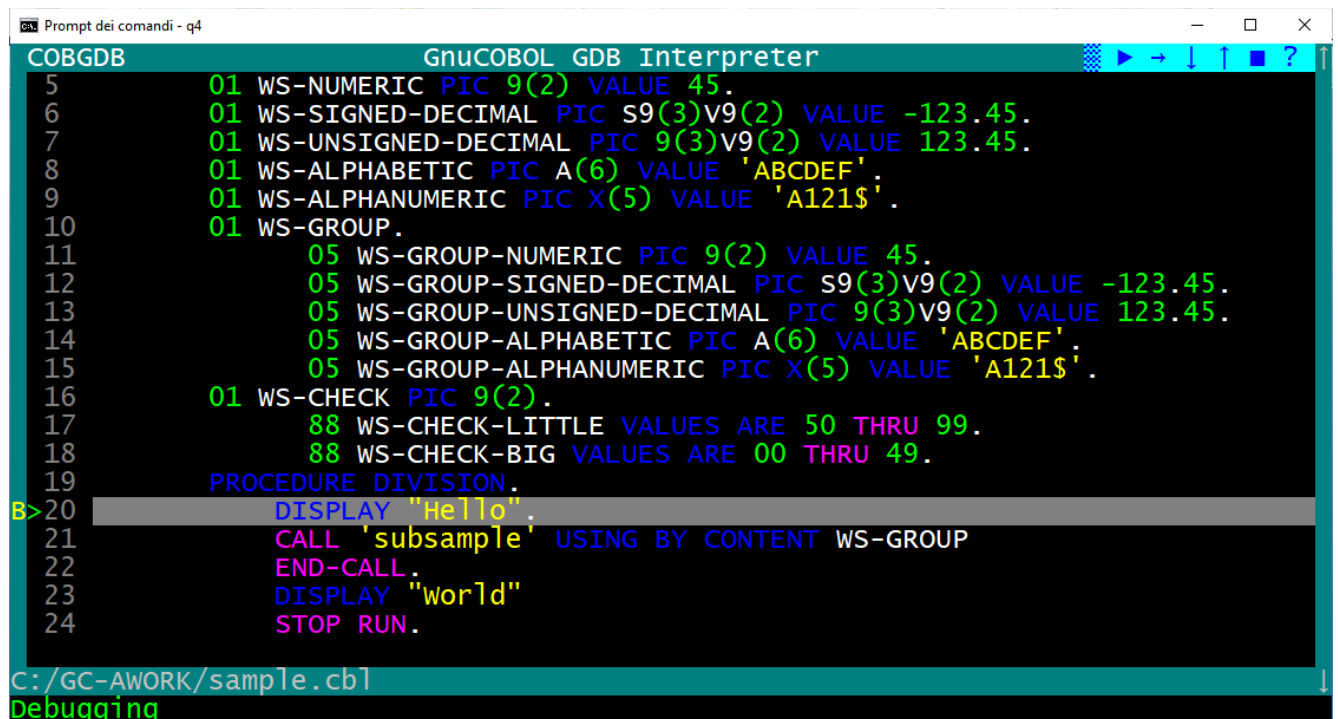
Source code is at <https://github.com/marcsosduma/cobgdb/tree/main/resources>.

This will create a single sample.exe executable.

This example shows that when you need to debug only subsample.cbl or only subsubsample.cbl you need to execute COBGDB with all three programs.

COBGDB sets the B breakpoint at first executable statement of first program "sample.cbl".

here use the R Run command to start the debugging session.




```

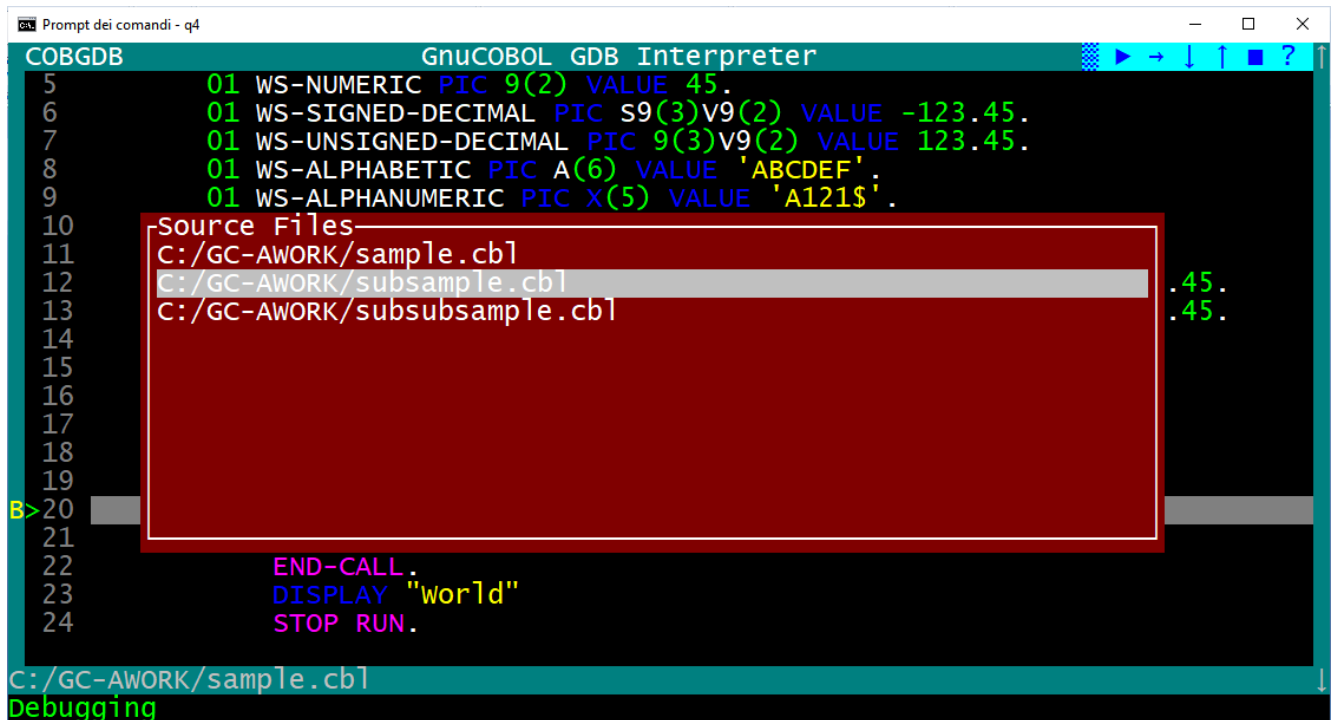
Prompt dei comandi - q4
COBGDB GnuCOBOL GDB Interpreter
5 01 WS-NUMERIC PIC 9(2) VALUE 45.
6 01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
7 01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
8 01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
9 01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10 01 WS-GROUP.
11     05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
12     05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
13     05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
14     05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
15     05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
16 01 WS-CHECK PIC 9(2).
17     88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
18     88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
19 PROCEDURE DIVISION.
B> 20 DISPLAY "Hello".
21 CALL 'subsample' USING BY CONTENT WS-GROUP
22 END-CALL.
23 DISPLAY "world"
24 STOP RUN.

C:/GC-AWORK/sample.cbl
Debugging

```

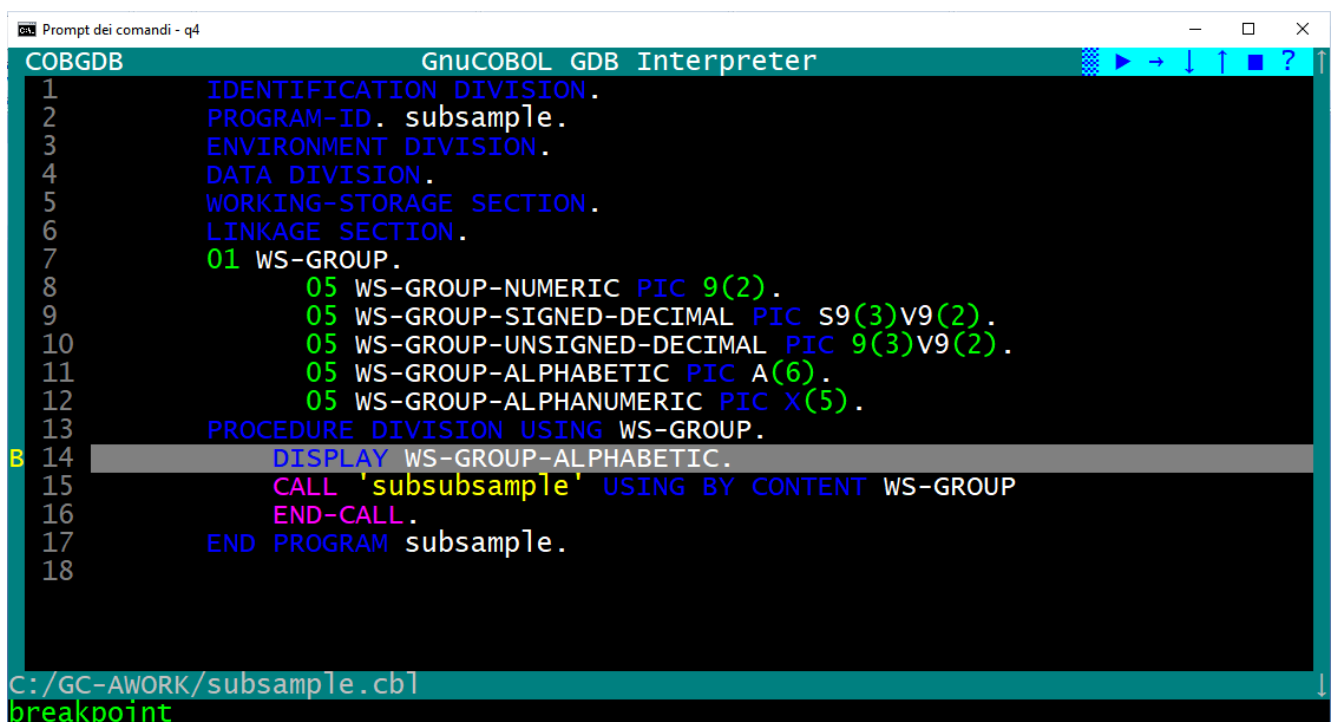
DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	26	

Now you can type the **F File** command and you will have the "Source Files" window.  
In this sample we select the second program in the list (subsample.cbl) and type Enter.




The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window. The 'Source Files' window is open, listing three files: C:/GC-AWORK/sample.cbl, C:/GC-AWORK/subsample.cbl (selected), and C:/GC-AWORK/subsubsample.cbl. The main window displays the source code of the selected file, subsample.cbl, with line numbers 5 through 24. The code includes data declarations for WS-NUMERIC, WS-SIGNED-DECIMAL, WS-UNSIGNED-DECIMAL, WS-ALPHABETIC, and WS-ALPHANUMERIC, followed by an END-CALL, DISPLAY "world", and STOP RUN statement. The status bar at the bottom indicates 'Debugging'.

COBGDB shows the selected program source code where in this sample we type a B command at line 14.



The screenshot shows the COBGDB GnuCOBOL GDB Interpreter window with the source code of subsample.cbl. The code includes an IDENTIFICATION DIVISION, PROGRAM-ID, ENVIRONMENT DIVISION, DATA DIVISION, WORKING-STORAGE SECTION, LINKAGE SECTION, and a PROCEDURE DIVISION USING WS-GROUP. The status bar at the bottom indicates 'breakpoint'.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	27	

now we type the F command again, then select the "sample.cbl" program and press Enter

```

COBGDB GnuCOBOL GDB Interpreter
1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. subsample.
3 ENVIRONMENT DIVISION.
4 DATA DIVISION.
5 WORKING-STORAGE SECTION.
6
7 Source Files
8 C:/GC-AWORK/subsample.cbl
9 C:/GC-AWORK/sample.cbl
10 C:/GC-AWORK/subsubsample.cbl
11
12
13
14 B
15
16
17
18
C:/GC-AWORK/subsample.cbl


```

now we are back to the sample.cbl program to continue the debugging session as we need.

```

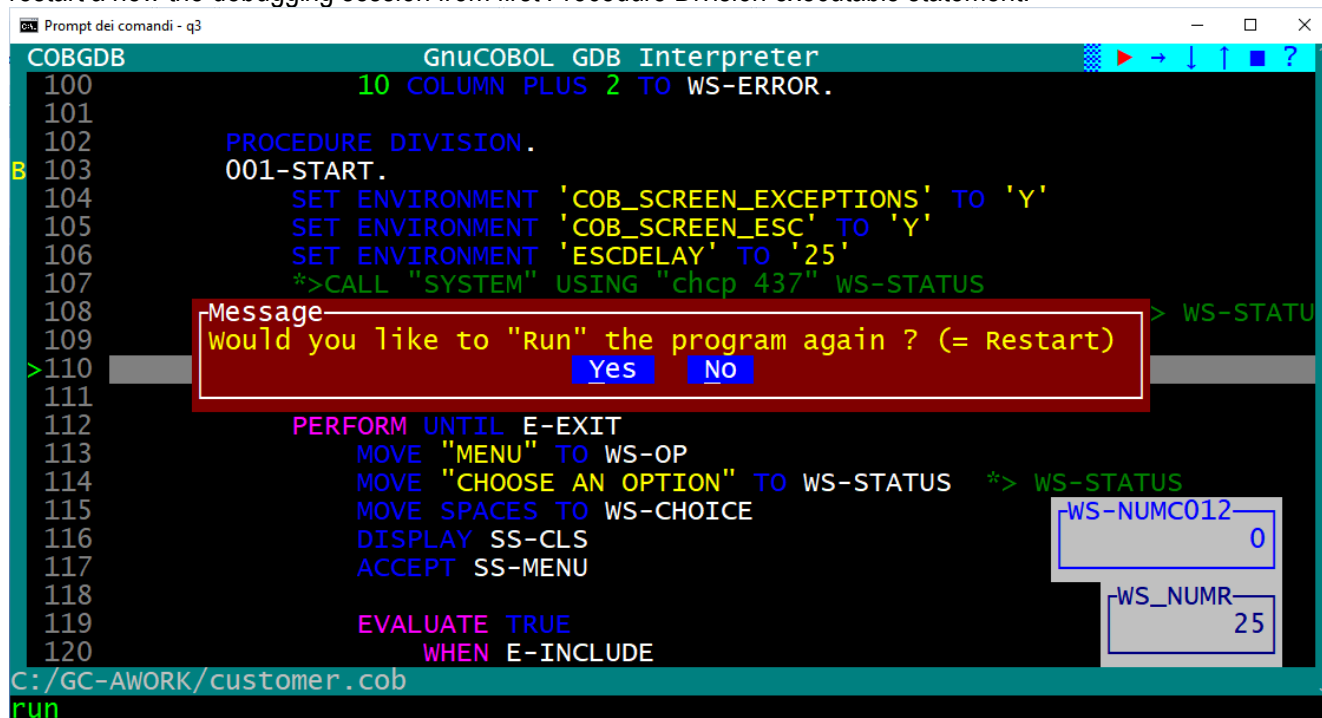
COBGDB GnuCOBOL GDB Interpreter
5 01 WS-NUMERIC PIC 9(2) VALUE 45.
6 01 WS-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
7 01 WS-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
8 01 WS-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
9 01 WS-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
10 01 WS-GROUP.
11 05 WS-GROUP-NUMERIC PIC 9(2) VALUE 45.
12 05 WS-GROUP-SIGNED-DECIMAL PIC S9(3)V9(2) VALUE -123.45.
13 05 WS-GROUP-UNSIGNED-DECIMAL PIC 9(3)V9(2) VALUE 123.45.
14 05 WS-GROUP-ALPHABETIC PIC A(6) VALUE 'ABCDEF'.
15 05 WS-GROUP-ALPHANUMERIC PIC X(5) VALUE 'A121$'.
16 01 WS-CHECK PIC 9(2).
17 88 WS-CHECK-LITTLE VALUES ARE 50 THRU 99.
18 88 WS-CHECK-BIG VALUES ARE 00 THRU 49.
19 PROCEDURE DIVISION.
20 B> DISPLAY "Hello".
21 CALL "subsample" USING BY CONTENT WS-GROUP
22 END-CALL.
23 DISPLAY "world"
24 STOP RUN.
C:/GC-AWORK/sample.cbl
Debugging

```

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	28	

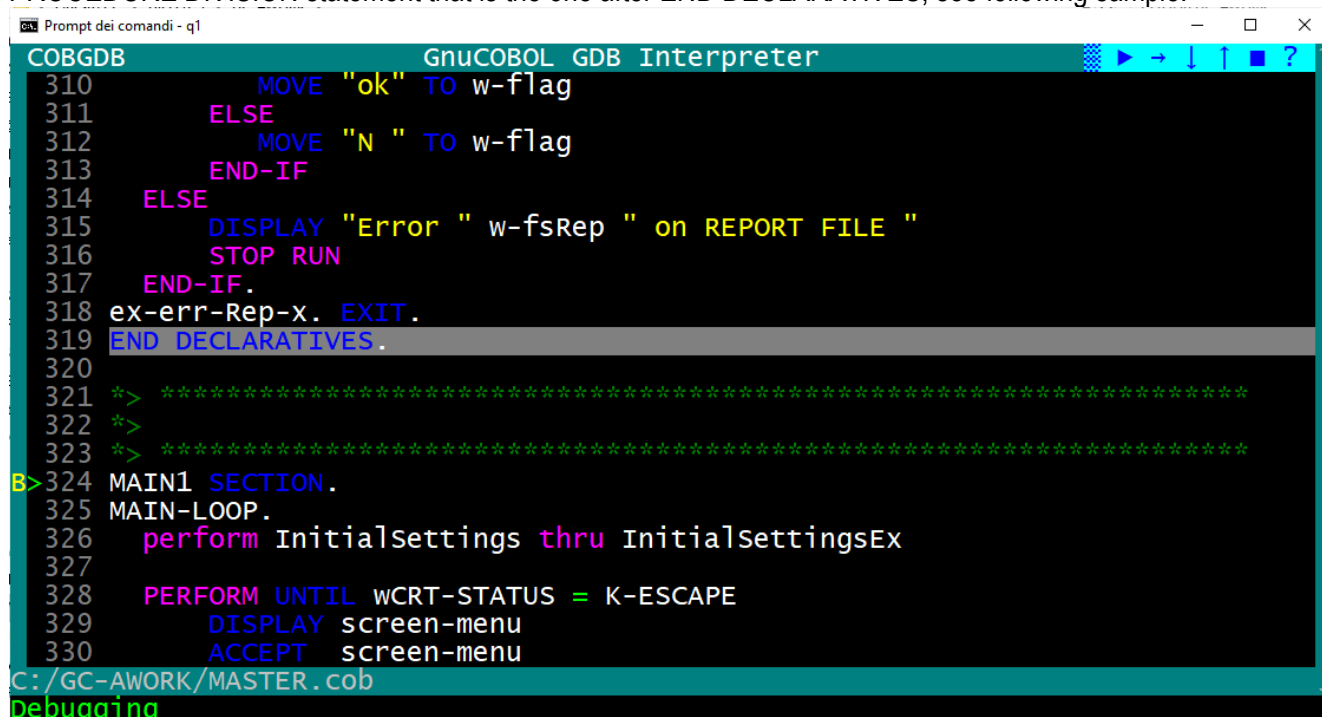
## 2.10. Run Command

If you click the Run command during a debug session you will receive a confirmation request, because Yes will restart a new the debugging session from first Procedure Division executable statement:




The screenshot shows the GnuCOBOL GDB Interpreter window. The code being debugged is in the Procedure Division, starting with '001-START.' and 'SET ENVIRONMENT' statements. A confirmation dialog box is displayed in the center, asking 'would you like to "Run" the program again ? (= Restart)' with 'Yes' and 'No' buttons. The 'Yes' button is highlighted. The dialog box also shows the current values of 'WS-NUMC012' (0) and 'WS\_NUMR' (25). The command prompt at the bottom shows 'C:/GC-AWORK/customer.cob' and the 'run' command has been entered.

Note: if program has DECLARATIVES then the first automatic B Breakpoint will be settled at first executable PROCEDURE DIVISION statement that is the one after END DECLARATIVES, see following sample:

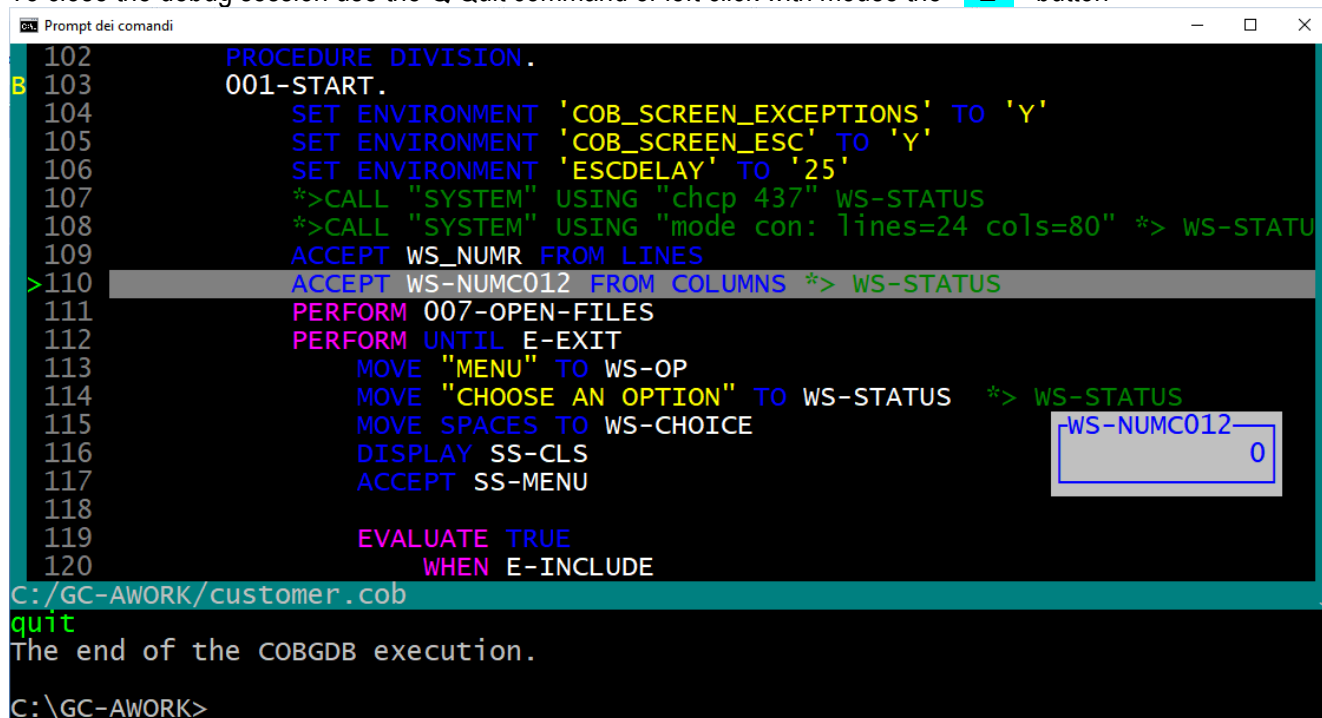


The screenshot shows the GnuCOBOL GDB Interpreter window. The code being debugged is in the Procedure Division, starting with 'ex-err-Rep-x. EXIT.' and 'END DECLARATIVES.' followed by a series of asterisks. The first automatic B Breakpoint is set at the first executable statement after 'END DECLARATIVES.', which is 'MAIN1 SECTION.'. The command prompt at the bottom shows 'C:/GC-AWORK/MASTER.cob' and the 'Debugging' command has been entered.

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	29	

## 2.11. Quit Command

To close the debug session use the **Q** Quit command or left click with mouse the  button




```

102      PROCEDURE DIVISION.
103      001-START.
104          SET ENVIRONMENT 'COB_SCREEN_EXCEPTIONS' TO 'Y'
105          SET ENVIRONMENT 'COB_SCREEN_ESC' TO 'Y'
106          SET ENVIRONMENT 'ESCDELAY' TO '25'
107          *>CALL "SYSTEM" USING "chcp 437" WS-STATUS
108          *>CALL "SYSTEM" USING "mode con: lines=24 cols=80" *> WS-STATUS
109          ACCEPT WS_NUMR FROM LINES
110      >110  ACCEPT WS-NUMC012 FROM COLUMNS *> WS-STATUS
111          PERFORM 007-OPEN-FILES
112          PERFORM UNTIL E-EXIT
113              MOVE "MENU" TO WS-OP
114              MOVE "CHOOSE AN OPTION" TO WS-STATUS  *> WS-STATUS
115              MOVE SPACES TO WS-CHOICE
116              DISPLAY SS-CLS
117              ACCEPT SS-MENU
118
119              EVALUATE TRUE
120                  WHEN E-INCLUDE

C:/GC-AWORK/customer.cob
quit
The end of the COBGDB execution.


C:\GC-AWORK>

```

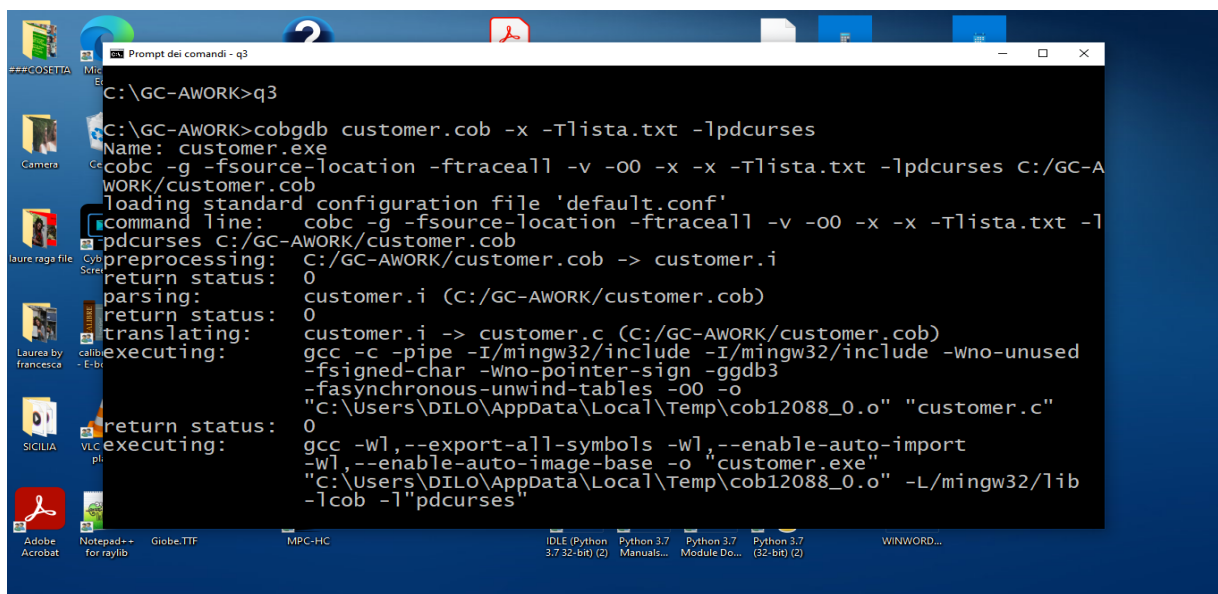
DOCUMENT CODE	MODULE: xxxxxxxxxx	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	30	

### 3. Document Change Log

CHANGE LOG
<p><b>Version1 of 2023.12.12.</b>  <i>First release</i></p> <p><b>. Version2 of 2023.12.23.</b>  <i>Step by Step sample of use is added</i>  <i>Some minor changes</i></p> <p><b>. Version3 of 2024.0218.</b>  <i>Restructured showing new cobgdb screens and features</i></p> <p><b>. Version4 of 2024.0401.</b>  <i>Allow variable modification when viewing the variable from a line of code</i></p>

DOCUMENT CODE	MODULE: XXXXXXXXXX	USING COBGDB FOR GnuCOBOL	PAGE	GnuCOBOL
GC-901	GC-XXXXXX	Author: Eugenio Di Lorenzo	31	

## Technical info



```

C:\GC-AWORK>q3
C:\GC-AWORK>cobgdb customer.cob -x -Tlista.txt -lpdcurses
Name: customer.exe
cobc -g -fsource-location -ftraceall -v -o0 -x -x -Tlista.txt -lpdcurses C:/GC-A
WORK/customer.cob
loading standard configuration file 'default.conf'
command line: cobc -g -fsource-location -ftraceall -v -o0 -x -x -Tlista.txt -l
pdcurses C:/GC-AWORK/customer.cob
preprocessing: C:/GC-AWORK/customer.cob -> customer.i
return status: 0
parsing: customer.i (C:/GC-AWORK/customer.cob)
return status: 0
translating: customer.i -> customer.c (C:/GC-AWORK/customer.cob)
executing: gcc -c -pipe -I/mingw32/include -I/mingw32/include -wno-unused
-fsigned-char -wno-pointer-sign -ggdb3
-fasynchronous-unwind-tables -o0 -o
"C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" "customer.c"
return status: 0
executing: gcc -w1,--export-all-symbols -w1,--enable-auto-import
-w1,--enable-auto-image-base -o "customer.exe"
"C:\Users\DILO\AppData\Local\Temp\cob12088_0.o" -L/mingw32/lib
-lcob -l"pdcurses"

```